

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Synthèse et propositions de spécifications théoriques d'un système de documentation

Pigneur, Y.

*Award date:*  
1977

*Awarding institution:*  
Université de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**Facultés Universitaires Notre-Dame de la Paix - Namur**  
**Institut d'Informatique**

**Année Académique 1976 - 1977**

118/8/54

**SYNTHESE et PROPOSITIONS**  
**de SPECIFICATIONS THEORIQUES**  
**d'un SYSTEME de DOCUMENTATION.**

Mémoire présenté en vue  
de l'obtention du grade  
de licencié et maître en  
informatique.

directeur: BOGAERT H.  
rapporteur: LESUISSE R.

**Y. PIGNEUR**

## INTRODUCTION

---

L'intention première de ce travail est de présenter les lignes de force et les concepts d'un système de documentation - pour l'étude ou l'élaboration des systèmes d'informations.

Comme point de départ de notre réflexion, nous avons admis que l'objectif général d'un tel système était de gérer une "base de données" ou stock documentaire

- fidèlement représentative du système d'informations en cours d'élaboration
- progressivement élaborée à partir des résultats obtenus au fur et à mesure du déroulement de la conception de ce système
- à la disposition des informaticiens et des utilisateurs en tant qu'élément de référence pour l'analyse, la construction et l'utilisation du système.

Vu l'ampleur d'un tel objectif, on comprendra que nous lui ayons apporté certaines restrictions, en particulier en ne prenant en considération que les activités de CONCEPTION des systèmes d'informations au détriment des activités de REALISATION - physique - de ceux-ci.

Ce sera l'objet de la première partie de préciser le contexte ou le cadre dans lequel s'insère un tel système de documentation et d'en fixer les limites mais aussi d'en présenter - dans ses grandes lignes - les spécifications générales.

Nous tenterons de montrer, dans cette partie, que tout système - automatisé - de documentation doit au minimum

- reposer sur un modèle (ou terminologie convenue), ensemble cohérent de concepts
- disposer d'un langage de manipulation de ces concepts; grâce auquel langage l'analyste pourra décrire la "réalité" tel qu'il la perçoit
- comporter un certain nombre d'outils pour enregistrer des renseignements dans le stock documentaire et interroger ce dernier, pour contrôler la cohérence interne et externe du système d'informations tel qu'il est perçu par l'analyste et pour en vérifier le caractère opérationnel.

Contrairement à une idée couramment répandue dans la littérature consultée, nous croyons - et essayerons de le justifier - qu'un système de documentation doit être le plus indépendant possible d'une quelconque méthode, dont nous ne contestons pas, par ailleurs, la nécessité.

D'autre part, ces mêmes spécifications serviront de grille d'analyse à la synthèse, que nous avons effectuée dans la seconde partie, de systèmes de documentation existants. La présentation - rapide - de ces systèmes à un double objectif : elle doit

non seulement être vue comme une illustration de la façon dont ces systèmes ont pris en charge ces spécifications - s'ils l'ont fait - et dans quelle mesure

mais aussi être considérée comme l'ensemble des "matériaux" à partir desquels nous avons élaboré une proposition de système qui constitue la troisième partie.



Dans cette dernière partie, nous n'aurons pas la prétention de proposer un outil - complet et opérationnel - de documentation tel que nous en aurons présenté les spécifications dans la première partie.

Nous nous limiterons à élaborer

- un modèle ou ensemble cohérent de concepts susceptibles de pouvoir prendre en considération toutes les "situations" qui risquent de se produire dans la réalité
- un langage qui permette de manipuler ces concepts pour décrire - et enregistrer dans le stock documentaire - une de ces "situations".

Le lecteur pourrait s'étonner de la présentation peu critique des 8 systèmes retenus dans la seconde partie et s'interroger sur l'intention logique qui a présidé au système proposé dans la troisième. Pour faire bref procès, retenons qu'il nous a paru opportun, après la présentation des spécifications générales (1ère partie), de proposer le schéma d'un système raisonnablement performant (3ème partie), en quelque sorte, induit de l'examen, pragmatique, de systèmes existants (2ème partie), étant ou ayant été opérationnels.



## TABLE DES MATIERES

	Introduction	1
	Table des matières	3
1.	CONCEPTION ET REALISATION DES SYSTEMES D'INFORMATIONS	5
1.1	POLES DE DESCRIPTION d'un SYSTEME D'INFORMATIONS	6
11.1	Fonctionnement du système	6
11.2	Structuration du système	9
11.3	Représentation du système	11
1.2	PROCESSUS D'ELABORATION d'un SYSTEME D'INFORMATIONS	13
12.1	Conception	14
12.2	Réalisation	16
1.3	SPECIFICATIONS GENERALES D'UN SYSTEME DE DOCUMENTATION	18
13.1	Limites	18
13.2	Objectifs	19
13.3	Composants	20
133.1	Modèle	
133.2	Langage	
133.3	Méthode	
133.4	Outils	
13.4	Coûts et bénéfices	23
2.	SYNTHESES DE SYSTEMES EXISTANTS	25
2.1	ADS	26
2.2	TAG	30
2.3	MINOS	33
2.4	DATA DICTIONARY	37
2.5	REMORA	40
2.6	CASCADE	44
2.7	CADIS	49
2.8	ISDOS	52
2.9	PROLONGEMENTS POSSIBLES...	57
29.1	... vers une automatisation	57
29.2	... vers une aide accrue au gestionnaire	59
3.	PROPOSITION D'UN SYSTEME DE DOCUMENTATION	60
3.1	MODELE CONVENU	61
31.1	Fonctionnement et structuration des traitements	62
311.1	Système	
311.2	Événement	
31.2	Structuration et Représentation des données	69
312.1	Entité (de l'organisation réelle)	
312.2	Information : concept de jonction	
312.3	Donnée : élément de représentation	
31.3	Identification des objets de notre modèle	80

3.2	LANGAGE PROPOSE	82
32.1	Création	83
32.2.	Suppression	85
32.3	Modification	85
32.4	Obtention	85
32.5	Complément : Macro	87
3.3	EXEMPLE D'UTILISATION	89
	Conclusions	93
	Remerciements	95
	Bibliographie consultée	96



# CONCEPTION et REALISATION des SYSTEMES d'INFORMATIONS.

---

Un tel titre aurait pu laisser présager que nous allions approfondir le processus d'élaboration d'un système d'informations. Malgré l'intérêt - mais aussi l'ampleur - d'une telle démarche, ce chapitre n'a pour seule prétention que de préciser le cadre (le contexte) dans lequel s'insère l'étude d'un outil de documentation.

Pour rencontrer cet objectif, nous serons amenés, dans un premier temps, à préciser les grands pôles de description d'un système d'informations dans une démarche de conception et d'analyse, abordant successivement le fonctionnement du système, la structuration de ses éléments et leur représentation.

Le second chapitre s'attachera, quant à lui, à dégager les principales activités ou phases qu'un processus d'élaboration d'un système d'informations - partiellement automatisé - exige. A ce niveau peu détaillé (ou macroscopique), cette découpe en phases devrait présenter une grande similitude avec celle rencontrée dans toute conduite d'un projet un peu ambitieux : ouvrage de génie civil, installation d'une nouvelle méthode de gestion, organisation d'une importante "manifestation"... Très schématiquement, la perception d'un besoin initialise un processus d'élaboration regroupant des phases de conception et de réalisation avant d'envisager la mise en exploitation du nouveau produit.

Nous terminerons cette partie introductive en présentant les spécifications générales d'un système de documentation : l'objectif qu'il se propose d'atteindre et les moyens dont il doit disposer. Nous estimons, en effet, que tout système d'aide à la conception doit comporter une terminologie convenue ou modèle, un langage pour traduire un phénomène réel dans ce modèle, une méthode de travail et des outils ou ensemble de programmes devant assurer des fonctions de documentation et, éventuellement, de contrôle.

Le système de documentation dont nous proposons les spécifications dans la suite de ce travail étant forcément limité et incomplet, cette première partie devrait notamment permettre de mieux visualiser la portée des restrictions faites à un tel système, en le limitant à certaines phases du processus d'élaboration et en ne prenant en considération que certaines composantes du système au détriment d'autres.



## 1.1

## POLES de DESCRIPTION d'un SYSTEME d'INFORMATIONS.

Une telle description devrait pouvoir se faire à partir d'une théorie du système d'informations. Mais dans la mesure où celle-ci est, à l'heure actuelle encore embryonnaire, nous nous contenterons de préciser les grands axes autour desquels s'articule la description d'un système d'informations reléguant à la seconde partie de ce travail (spécifications de systèmes existants) le soin de préciser les concepts que proposent les systèmes existants pour en tenir compte.

Ainsi au même titre qu'un ingénieur décrit le fonctionnement d'une nouvelle installation, en élabore les plans et le cahier des charges, c'est-à-dire mise en oeuvre, assemblage des matériaux et descriptions de ceux-ci, le "concepteur" ou "l'analyste" examine les éléments caractéristiques d'un système d'informations sous la triple approche du fonctionnement de celui-ci, de la structuration de ses composantes et de leur représentation.

## 11.1

### FONCTIONNEMENT du SYSTEME

#### ORGANISATION ET SYSTEME D'INFORMATIONS:

Peu désireux d'engager une polémique à propos de terminologies, nous avons emprunté à Langefors sa définition d'un système d'information qui nous semblait suffisamment générale pour recouvrir la plupart des acceptions communément admises dans la littérature.

Un système d'informations est utilisé pour fournir et traiter de l'information -quel que soit l'usage qui en est fait.  
(Langefors, 1974)

Complétons cette définition par une observation :

Un système d'informations est toujours destiné à rendre des services ou exercer un contrôle sur un autre système : son système objet (ou organisation).  
(Langefors, 1974).

Un élément important dans cette vision est que le système d'informations doit, ou devrait, être désigné pour servir le système de gestion.

(Teichroew, Sayani, 1971).

En faisant nôtres ces 2 remarques, nous voulons insister sur la nécessité de situer l'étude d'un système d'informations dans le contexte d'une organisation réelle. Et une façon de veiller à l'adéquation de ce système d'informations-automatisé ou non-aux exigences de l'organisation et de ses utilisateurs est de prendre comme prémices d'une démarche de description l'étude des flux d'informations.



## UNITES-COMPOSANTES ET FLUX DE RESSOURCES:

L'"information", à ce niveau, doit, en effet, être vue comme une ressource, au statut peut-être particulier, parmi les nombreuses ressources (humaines, physiques, monétaires...) qu'une organisation intègre et coordonne en vue de la transformation de matières en produits ou de la prestation de services.

Ces ressources circulent donc, à des fréquences plus ou moins déterminées, entre l'organisation et l'environnement, mais aussi d'une unité-composante de l'organisation (= cellule homogène de traitement de l'information) à l'autre.

Aussi, pour obtenir une image plus détaillée de l'organisation, utile à l'élaboration de son système d'informations, convient-il non seulement d'identifier les ressources et leur flux, mais aussi les unités-composantes qui émettent, reçoivent et transforment ces ressources.

Il est clair que les ressources informationnelles et les flux associés seront privilégiés, mais il n'est pas absurde d'imaginer qu'une démarche de conception prenne en considération, dans une certaine mesure, les flux d'autres ressources. Citons, à ce propos, la méthode SOP (IBM, 1961) et son document "Ressources Usage Sheet".

Malgré la divergence de terminologies et de modalités de mise en oeuvre, la plupart des auteurs reconnaissent l'importance de l'identification des unités-composantes et des flux de ressources informationnelles :

Il s'agit dans un premier temps de procéder à l'inventaire des postes-émetteurs ou récepteurs de documents - et les documents-émis ou reçus par ces postes.

(Bourgeois, 1977).

Toutes les informations rassemblées proviennent de flux d'informations qui se réalisent entre les différentes cellules d'activités.

(Bodart, 1976).

Les centres de travail et les secteurs d'environnement sont identifiés (...) Un flux d'informations fait allusion à tout transfert, à intervalle régulier ou non, d'informations.

(Orden, 1972).

Pour déterminer les informations que le système d'informations doit fournir, nous devons définir les fonctions et les sous-systèmes du système-objet que manipuleront ces informations.

(Langefors, 1974).

En général, la description d'un système d'informations, outre les informations et leur traitement, inclut les unités d'organisation qui reçoivent (transforment et) émettent ces informations.

(Teichroew, Rataj, Hershey, 1975).



A un niveau d'observation différent, une unité-composante de l'organisation présente les mêmes caractéristiques externes que l'organisation globale puisqu'elle intègre (réception, transformation et émissions) également des ressources pour réaliser un objectif déterminé. La définition d'une "cellule d'activités" est, à cet égard, assez significative :

| Une cellule d'activités est un centre d'activités homogènes, doté de ressources et pourvu de règles de comportement nécessaires à son fonctionnement.

(Bodart, 1976).

On aurait pu donner la même définition d'une organisation ou d'un département !

Enfin, il est important d'insister sur l'aspect dynamique de ces flux puisque ces émissions, ces transformations et ces réceptions de ressources informationnelles au sein des unités-composantes ne se font qu'à certains moments dont la fréquence est plus ou moins fixée. Par ailleurs, les volumes de ces transferts de ressources peuvent également être sujets à de plus ou moins grandes variations. Cet aspect dynamique est d'autant plus tangible si l'on songe à des unités-composantes temporaires par rapport à la vie de l'organisation.

#### ACTIVITES OU PROCESSUS DE TRAITEMENT:

L'identification et la description du processus de traitement des ressources informationnelles au sein des unités-composantes de l'organisation constituent l'une des tâches principales de la démarche de conception; à tel enseigne qu'elle fut - trop - souvent considérée comme le seul objectif de l'analyse fonctionnelle classique.

Notons, au risque de dérouter certains, la similitude de nature d'une unité-composante de l'organisation et d'un processus de traitement puisque tous les 2 reçoivent, coordonnent et transforment des ressources - notamment informationnelles - pour produire des résultats qui expriment leur objectif ou fonction. Ainsi, même pour un processus en Operating system

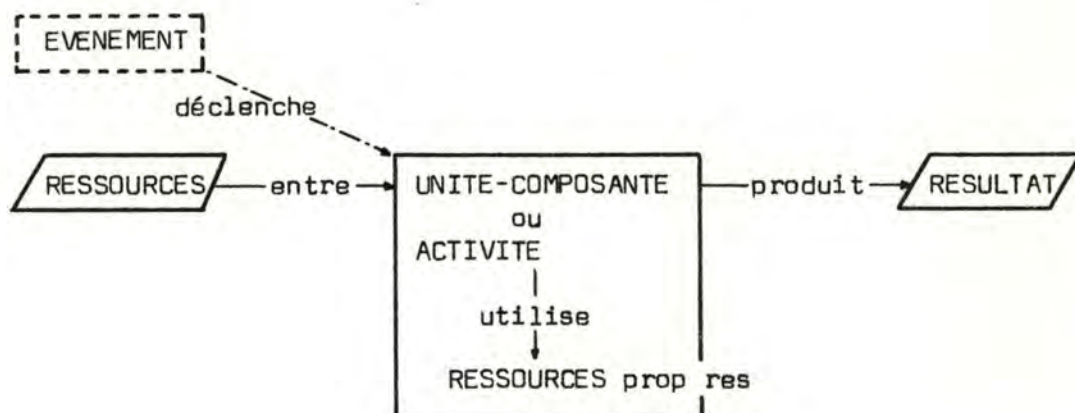
| Pour qu'il puisse évoluer, il a besoin de procédures et de données, de mémoire, d'unité-centrale, de fichiers, de périphériques. Nous appelons ces entités des ressources.

(Crocus, 1975).

Dans une vision dynamique, il ne faut pas négliger l'évolution de ces activités ou processus de traitement, dans le temps et il importe d'examiner les événements qui déclenchent leur mise en oeuvre.



Schématiquement, le fonctionnement du système se résume à :



## 11.2

### STRUCTURATION du SYSTEME

Pour rendre compte du fonctionnement du système, les éléments intervenants ont été logiquement inventoriés sous les qualifications de "unités-composantes", "processus de traitement", "ressources", "événements" comme on l'a vu ci-dessus.

#### UNE DEUXIEME APPROCHE DE LA MEME REALITE:

Effectivement, cette première approche de fonctionnement ne décrivait pas les éléments intervenants, dans leurs caractéristiques d'état, de hiérarchie ou d'"architecture". Une image correcte de leurs relations réciproques, de leur décomposition et/ou de leur assemblage impose donc la présente approche de structuration.

Il est, en effet, indispensable d'étudier les rouages internes de la "machine" à mettre en place. Si, donc, l'étude des relations existantes entre données et traitements a fait l'objet de l'approche de fonctionnement, il convient maintenant de s'attacher aux relations liant les données entre elles et les traitements entre eux.

#### TRAITEMENTS ET DONNEES:

Car cet effort de structuration doit, bien sûr, porter sur les données et les traitements. Ainsi, par exemple, Teichroew prévoit-il explicitement que la description d'un système d'informations contienne notamment :

- Une définition des données par structuration, contenant des informations sur la façon dont les éléments associés aux inputs et outputs sont structurés et reliés.
- Une définition des processus par structuration, reprenant la hiérarchie des processus qui constituent le système et définis au niveau de détail souhaité.

(Teichroew, Rataj, Hershey, 1974).



Mais c'est surtout au niveau des "données mémorisées" ou "ressources propres" que la nécessité de cette structuration est particulièrement tangible et justifie les nombreuses recherches en conception de bases de données.

#### DECOMPOSITION ET SYNTHESE:

Cette structuration intervient effectivement dans deux situations distinctes rencontrées dans la conception ou l'analyse d'un système d'informations :

- soit lors d'une DECOMPOSITION cognitive de la réalité observée (décomposition progressive en éléments constitutifs, jusqu'au niveau des données ou des opérations élémentaires). Ainsi, un exemple parmi d'autres, en ce qui concerne les données :

Les "pro-concepts" sont progressivement définis par d'autres "pro-concepts" plus fins et éventuellement par des "elementary concepts", à travers une série d'analyse de décomposition...

(Langefors, 1974).

- soit lors d'une SYNTHESE reconstitutive du système correspondant. Langefors, par exemple, continue ainsi :

...Quand cette analyse de décomposition est terminée(...), il convient parfois de reconstituer des "concepts consolidés", conglomérats d'"elementary concepts".

(Langefors, 1974).

C'est après avoir effectué ces différentes "synthèses" au niveau des traitements et des données (en fonction notamment des accès aux données mémorisées = fonctionnement) que l'on disposera de l'image structurée du système qu'il conviendra de physiquement réaliser ou REPRESENTER (Cfr. infra) par l'installation d'une "base de données" et par la réalisation d'une bibliothèque de programmes.

Par ailleurs, cette nécessité de structuration justifie notamment la recherche de nomenclature standard des traitements ou des données pour fixer les règles de décomposition ou de synthèse. Par exemple :

Cette nomenclature exprime une hiérarchisation des éléments de la structure d'un système d'informations (...). Nous distinguons la sous-structure des données (donnée, information élémentaire, unité d'information, structure logique, structure fonctionnelle). et celle des traitements (fonction, phase, application, sous-système et système).

(Bodart, 1976).



## "STRUCTURE CONCEPTUELLE"

Dans un affinement, second dans notre démarche, il semble intéressant de se fixer des critères de STRUCTURATION - en particulier des données mémorisées - qui permettraient de répondre aux besoins de stabilité et d'autonomie des descriptions d'éléments, indépendamment de leur mise en oeuvre dans tel ou tel combinaison "opérationnelle".

C'est ce qui a incité certains à proposer une "structure conceptuelle" amenant à étudier les phénomènes en termes d'"objets", de "relations" entre objets et de "propriétés" des objets et des relations.

Ce modus operandi est familier, depuis quelques temps déjà, aux chercheurs en conceptions de base de données dont on connaît le souci de polyvalence et de généralité :

Nous assurons que l'expérience du monde réel amène les gens à reconnaître que ce monde réel peut être complètement décrit par des entités, des associations entre ces entités et leurs propriétés.

(Benci, Bodart, Bogaert, Cabanes, 1976)

Il est admis que la description du système objet repose sur les concepts de "objet", de "relations d'objets" et de "propriétés".

(Sundgren, 1974).

Notons, enfin, que le travail de structuration doit également être envisagé au niveau des moyens mis en oeuvre pour que le système d'informations fonctionne :

Structure hardware et software (machine) nécessaire et structure de l'organisation (homme) à mettre en place.

### REMARQUE:

On comprendra qu'il n'existe pas forcément de relation d'antériorité obligée de l'approche de fonctionnement par rapport à celle de structuration. La connaissance de la réalité observée, sa compréhension et son affinement progressif impliquera très souvent un mouvement alternatif de va-et-vient entre les 2 approches évoquées.

"Pôles" ne signifient pas "étapes"!

## 11.3

### REPRESENTATION du SYSTEME

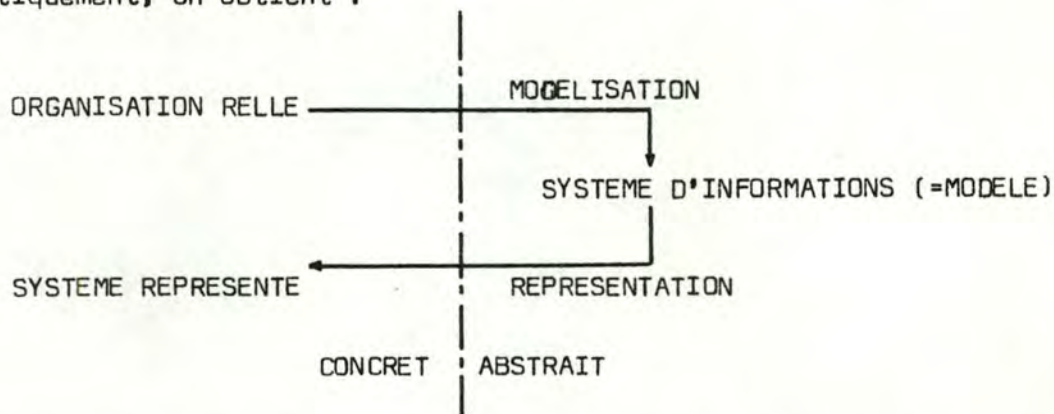
Dans son effort pour rendre compte de la réalité observée, l'analyste utilise donc les schémas ABSTRAITS de l'approche de fonctionnement et de celle de structuration, visant à décrire les caractéristiques qualitatives des éléments observés. Il convient cependant qu'il exprime concrètement le résultat de sa construction et qu'il fasse donc retour à une expression pratique, communicable et susceptible d'être "enregistrée". En d'autres termes, il lui faudra également retenir la "REPRESENTATION" donnée et à donner aux éléments sur lesquels a porté son étude ou comme le dit Langefors :



Pour "implémenter" un système d'informations à travers un système de données, il faut établir comment les données enregistrées peuvent être utilisées pour représenter les "informations" à propos des phénomènes du monde réel.

(Langefors, 1974).

Schématiquement, on obtient :



Nous empruntons à (Langefors, Sundgren, 1975) les deux raisons fondamentales pour justifier cette découpe (organisation réelle, modèle du système d'informations, système représenté-réel); découpe que la plupart des chercheurs en conception de bases de données s'entendent pour juger indispensable.

La distinction entre système modelisé (information) et système représenté (données enregistrées) qui, dans une vision rapide, aurait pu paraître subtile et même superfétatoire est cependant intéressante et justifiée à plus d'un titre.

Car, d'une part, affirmer que les données "pointent" ou "font référence" directement aux objets ou phénomènes du monde réel, c'est négliger la conception que l'homme a de cette réalité, or cette conception est un véritable tampon entre la réalité et les données enregistrées.

D'autre part, méthodologiquement cette différenciation permet, dans un premier temps, à l'utilisateur de définir ses besoins en informations dans un langage qui lui soit accessible et, par contre, laisse toute liberté ultérieure aux informaticiens pour le choix et le maintien de la "représentation" la mieux adaptée (notamment aux moyens technologiques) de la structure (modèle) des données et des traitements pour tenir compte de ces besoins.



## **PROCESSUS d'ELABORATION d'un SYSTEME d'INFORMATIONS.**

---

Toute élaboration d'un système d'informations - automatisé - est un projet qui nécessite la prise en charge d'un certain nombre d'activités fondamentalement identiques, même si elles peuvent être organisées ou abordées de nombreuses manières.

Il n'entre pas dans les intentions de ce travail de répertorier et de détailler ces différentes "organisations" possibles, ni même d'en privilégier une par rapport aux autres; nous nous contenterons simplement d'évoquer les principales activités - dans un ordre chronologique - d'un pareil processus d'élaboration.

Il faut souligner que le principal mérite d'une découpe de ce processus en étapes ou ensembles d'activités clairement différenciés et circonscrits est de pouvoir rendre une solution, obtenue au terme d'une étape, indépendante des choix qui seront faits dans les étapes ultérieures. Cette particularité sera particulièrement exploitée par des systèmes se proposant d'automatiser complètement ce processus d'élaboration (ISDOS, par exemple, Cfr. 2ème partie).

Ainsi, avant de songer à la mise en exploitation d'un système dont on a PERCU LE BESOIN, il sera avantageux, sinon indispensable, de clairement dissocier les activités de CONCEPTION des activités de REALISATION du système.

En fait, ce distinguo est repris comme tel par la plupart des auteurs et pour n'enciter que deux

| Le processus de construction d'un système d'informations peut être divisé en deux étapes.

| Celle de définition  
| et celle de réalisation.

(Bukenko, Kallhamar, 1973).

| Nous différencions les deux types de problèmes.

| Quelles sont les informations nécessaires pour répondre aux besoins de l'utilisateur ?

| Comme le système est-il organisé et opère-t-il pour exploiter au mieux la technologie ?

(Langefors, 1974).

### **PERCEPTION d'un BESOIN**

Ce processus d'élaboration d'un système d'informations est effectivement déclenché ou initialisé par la PERCEPTION D'UN BESOIN (°) par un utilisateur; signalons cependant que rien n'empêche de pla nifier ces besoins au sein, par exemple, d'un plan directeur.

---

(°) Dans notre esprit, la PERCEPTION DES BESOINS est moins une activité ou une phase que le mécanisme qui déclenche le processus d'élaboration d'un système d'informations.



Et selon la nature de ce besoin, il en résultera 2 grands types d'approches possibles amenant :

- soit la création d'un nouveau système pour faire face à une situation de non-existence
- soit la modification d'un système existant en vue
  - de le perfectionner dans la mesure où il donne satisfaction à ses utilisateurs
  - ou d'améliorer ce système existant mais non satisfaisant.

C'est donc à ce niveau que l'utilisateur formule sa toute première vision des choses et qu'il confie éventuellement ses desiderata et le soin d'élaborer une solution aux personnes chargées de concevoir et de réaliser un système en vue de sa future mise en exploitation.

## 12.1

### CONCEPTION

On regroupe généralement sous ce vocable les activités correspondant à :

- une préétude du projet,
- une étude du système d'informations,
- et, éventuellement, la réalisation d'une maquette ou d'un prototype.

Cette phase de conception débute, en effet, le plus souvent par une - **préétude** destinée à définir plus précisément le problème, à examiner son opportunité et à déterminer les spécifications de la solution à élaborer ne serait-ce que pour estimer les coûts et les bénéfices de l'opération. C'est généralement la rédaction d'un cahier des charges qui clôture cette activité.

Après les approbations d'usage, le projet rentre alors dans une phase d'

- **étude** - plus souvent appelée de conception logique ou d'analyse fonctionnelle - où l'on est amené à déterminer le contenu du système d'informations de la façon la plus indépendante possible de la technologie, de l'équipement disponible et même de toute "représentation" de ce système au moyen de programmes et de données enregistrées.

Cette indépendance vis-à-vis de la technique informatique est, à l'heure actuelle, exprimée par la plupart des auteurs :

L'étape conceptuelle est l'étape de définition des problèmes dans une forme aussi indépendante que possible des moyens qui permettront de les traiter automatiquement ou non.

(Rolland, 1977).

Pendant l'étape de définition, les propriétés externes et internes du système d'informations sont définies indépendamment de l'ordinateur.

(Bukenko, Kallhamar, 1973).

Il est important de définir les besoins en information et seulement après de considérer les données (et leurs combinaisons) pour les représenter.

(Langefors, 1974).



Cette indépendance est, bien sûr, toute relative !  
Et s'il faut tenter de minimiser les feed-back des phases de réalisation vers celles de conception, il serait vain d'espérer les supprimer. Il n'est d'ailleurs pas certain que cette indépendance totale - idéale - n'ait que des avantages.

A cet objectif d'indépendance, il faut donc associer la recommandation d'éviter au maximum que les décisions prises lors des activités de conception ne soient remises en question dans les phases ultérieures de réalisation. (°)

Cette phase, essentiellement caractérisée par une collaboration prononcée de l'utilisateur du système et l'analyste doit donc veiller

- à décrire le fonctionnement du système dans ses aspects statiques et dynamiques
- à proposer une première structure des traitements et des données
- à rassembler les quelques renseignements utiles à la représentation du système et connus de l'utilisateur.

Ainsi pour Teichroew

La conception logique, c'est le moment où  
l'on détermine les informations nécessaires pour résoudre le problème  
l'on conçoit une organisation générale du système d'informations et l'on décrit ces informations avec un détail suffisant et d'une manière qui convienne à la fois à l'utilisateur et aux analystes.

(Lesuisse, 1976) citant (Teichroew, Rataj, Hershey, 1975)

Dans certains gros projets, cette ETUDE pourra avantageusement être prolongée par la réalisation et la présentation d'une

**maquette** destinée à valider la solution proposée et retenue avant de la réaliser définitivement.

F. Peccoud, concepteur d'un tel projet (MACSI-P), précise de la façon suivant l'enjeu de la réalisation d'un prototype

Un prototype permet un nouveau processus de développement dans lequel puisse s'exprimer avant la réalisation de la version opérationnelle, la plupart des critiques qu'appelle sa conception.

(Peccoud, Giraudin, 1977).

Un prototype doit donc être "fonctionnellement équivalent" au système dont il n'est que la maquette : mêmes entrées, mêmes sorties et même logique de traitement; par contre, les moyens mis en oeuvre pour les réaliser différeront sensiblement. Compte tenu de ces remarques, les moyens indispensables pour élaborer un prototype doivent permettre, selon F. Peccoud :

(°) Si, à l'heure actuelle, seuls le bon sens et l'expérience permettent de concilier ces deux objectifs apparemment antinomiques, il n'est pas absurde d'imaginer des recherches ultérieures qui répondent à ce genre de préoccupations d'une façon plus péremptoire et argumentée (en fonction de critères d'appréciation à mettre en évidence).



- une réalisation rapide
- une réalisation facilement modifiable
- une autodocumentation maximale du prototype dans la mesure où la dernière version de celui-ci servira de cahier des charges pour la réalisation de la version opérationnelle du projet.

Au sortir de cette phase de CONCEPTION on devrait donc disposer

d'une image conceptuelle, complète, cohérente et non ambiguë des phénomènes réels pris en compte.

(Rolland, 1977).

C'est ce schéma logique du système en construction qu'il va falloir REALISER (ou "implémenter") à partir des moyens (humains et techniques) que l'on se donne.

## 12.2

### REALISATION

Voici, à titre d'exemple assez significatif, comment deux auteurs perçoivent cette phase :

Réaliser, c'est créer les moyens, les mettre en place et les rendre aptes à fonctionner.

(Rolland, 1977).

La réalisation implique que nous devons créer un système physique qui prenne en compte les propriétés structurelles, fonctionnelles et dynamiques définies précédemment.

(Bukenko, Kallhamar, 1973).

Cette phase devra donc s'attacher à

- étudier les moyens à mettre en oeuvre
- proposer une architecture physique du système et
- construire et tester la version opérationnelle de celle-ci.

**l'étude des moyens** consiste

- éventuellement à choisir une technologie et un équipement appropriés aux spécifications logiques.
- plus souvent à déterminer les ressources informatiques (hardware et software - nécessaires et à adapter les ressources dont on dispose aux nouveaux besoins.
- presque toujours à préciser les moyens humains dont on aura besoin pour l'exploitation du nouveau système (responsables des saisies et de l'alimentation du système en informations, destinataires des résultats...).

C'est effectivement sur base des spécifications logiques du système et partiellement à partir de l'étude précédente que l'on pourra étudier et proposer

une **architecture physique** du système d'informations. Il s'agit donc d'une adaptation du schéma logique aux moyens dont on dispose et il doit en sortir

- une organisation des traitements et des données
- et une proposition de planning ou scheduling des traitements pour rencontrer les exigences des utilisateurs en temps voulu.



Teichroew regroupe les deux activités précédentes sous l'appellation "conception physique du système" qu'il définit ainsi :

- En fonction des besoins exprimés, on se propose de choisir
- l'organisation du traitement (Batch, Temps réel,...)
  - et le hardware nécessaire
- et de déterminer
- une structure optimum des programmes
  - et une structure optimum de la base de données.

(Lesuisse, 1976) citant (Teichroew, 1971).

Il restera alors à réellement

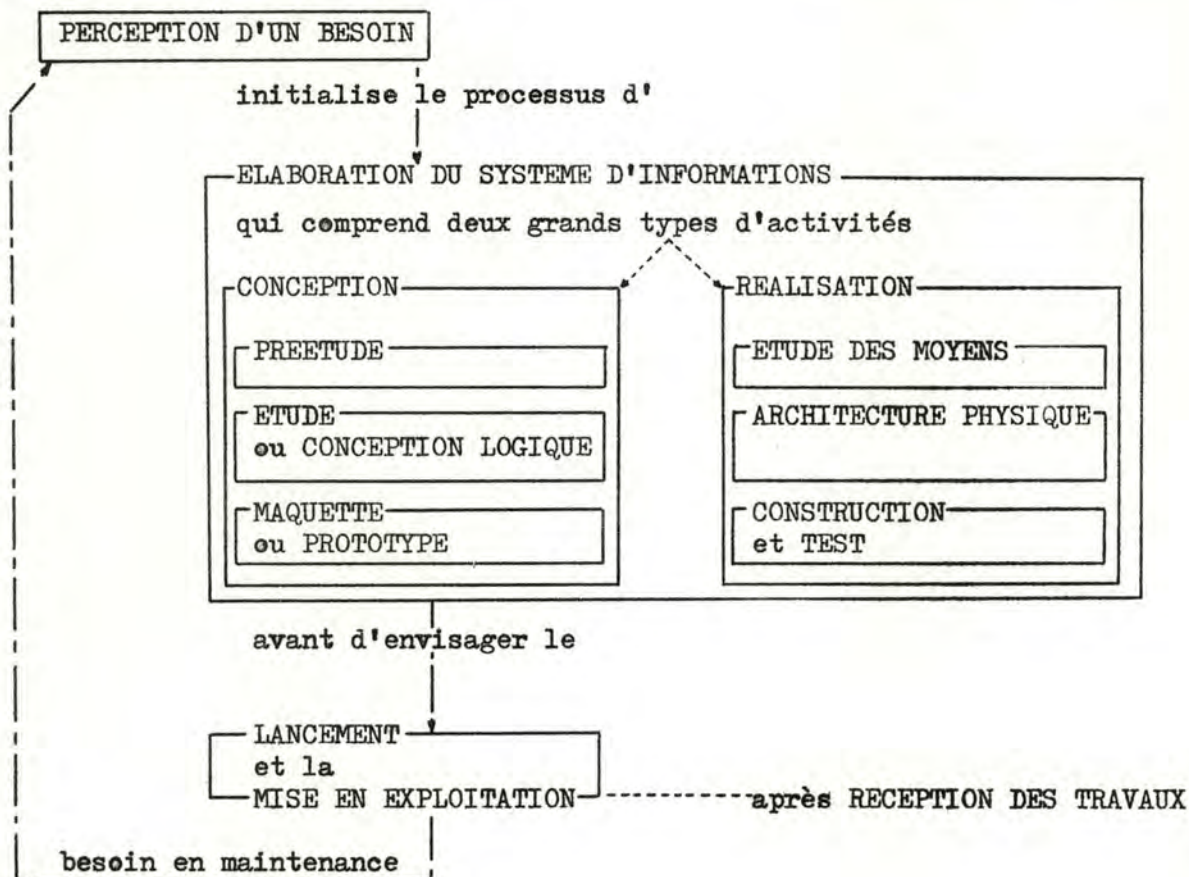
**construire / tester** le système, c'est-à-dire programmer et construire les fichiers ou la base de données et achever ainsi le travail de "représentation" amorcé lors de la phase précédente.

C'est seulement après avoir testé le système que l'on pourra envisager le **LANCEMENT**, la "Réception des Travaux" par l'utilisateur et la **mise en exploitation** "opérationnelle" du nouveau système.

REMARQUE :

La présentation qui vient d'être faite ne doit pas laisser croire à un processus linéaire des activités à réaliser, l'expérience est là pour rappeler qu'il s'agit d'un processus ITERATIF et dans lequel on peut trouver plusieurs tâches à différents niveaux d'achèvement.

Le mérite d'une bonne méthode d'analyse ou d'une bonne organisation de ces activités est précisément de réduire le nombre de ces itérations.





## **SPECIFICATIONS GENERALES d'un SYSTEME de DOCUMENTATION.**

(°)

Chacune des phases ou ensembles d'activités que nous venons très brièvement de présenter génère de la documentation à propos du système à élaborer. Cette documentation a principalement pour rôle de

- décrire le système d'informations - dans son état d'avancement
- faciliter le développement de ce système
- mais aussi faciliter la gestion de celui-ci après sa mise en oeuvre ou comme le signale Teichroew :

Cette documentation est nécessaire pour enregistrer les résultats des décisions qui seront nécessaires plus tard, pour communiquer entre les différents individus collaborant au projet, pour donner les instructions aux personnes qui devront employer ce système et pour disposer d'une base pour les modifications éventuelles du système.

(Teichroew, Rataj, Hershey, 1975).

Si une telle documentation gérée manuellement peut apporter une aide substantielle aux analystes et aux utilisateurs, les avantages d'une gestion par ordinateur de cette documentation apparaissent clairement lorsque la taille des systèmes à élaborer croît et que les modifications de ceux-ci sont fréquentes. Ceci nécessite alors l'installation d'une base de données - pour remplacer les dossiers et autres documents manuels - et des moyens - notamment un langage - approprié pour la mettre à jour et pour la consulter.

Le but de ce chapitre est donc de présenter les spécifications générales, objectif et composantes, d'un système - automatisé - de documentation que nous aurons préalablement limité à certaines activités du processus d'élaboration d'un système d'informations. A ce propos, nous en sommes venus à considérer que tout système de documentation, ou plus généralement, tout système d'aide à la conception devait, au minimum, comprendre

- un MODELE
- un LANGUAGE
- un certain nombre d'OUTILS devant assurer certaines fonctions
- et une METHODE de travail. Toutefois, nous avons la conviction, et nous essayerons de la justifier, qu'un outil de documentation doit être suffisamment indépendant de toute méthode au risque d'une trop grande rigidité et donc d'une certaine inefficience.

Nous terminerons ce chapitre en mentionnant les coûts et les bénéfices qu'on est en droit d'attendre d'un tel système de documentation.

### **LIMITES**

Toute "documentation" pour être complète doit comporter des renseignements concernant

- non seulement le système à élaborer
- mais aussi le projet d'élaboration lui-même, sa gestion et son avancement.

---

(°) Ce chapitre est principalement élaboré à partir de l'article de (Teichroew, Rataj, Hershey, 1975) et de l'étude que nous avons effectuée (2ème partie) de Systèmes existants.



Quoique parfaitement conscient de l'importance d'une telle documentation et de l'intérêt à son automatisation partielle, (project management), spécialement pour les gros projets, nous ne l'avons pas prise en considération dans ce travail.

D'autre part, envisager un système - automatisé - de documentation complet, prenant en compte toutes les activités du processus d'élaboration d'un système d'informations, depuis la perception d'un besoin jusqu'au lancement en exploitation, nous a paru présomptueux dans le cadre de ce travail. Aussi nous sommes-nous limités à étudier et proposer les spécifications d'un système de "documentation des exigences de l'utilisateur" et donc uniquement destiné à enregistrer les renseignements obtenus lors de la conception du système.

## 13.2

### OBJECTIFS

Dans la mesure où, à ce stade, le principal souci est d'identifier les besoins en informations (quoi?) pour répondre aux exigences de l'utilisateur laissant de côté la façon technique (comment?) d'atteindre le résultat escompté, le système de documentation correspondant devra donc

- en cours d'analyse servir de moyen de communication et d'outil d'analyse en restituant une image toujours à jour du système en construction
- en fin de parcours constituer l'image-conceptuelle - complète, cohérente et non ambiguë du futur système d'informations tel que le voit l'organisation et tel qu'il sera construit dans les étapes ultérieures.

Ces objectifs sont, bien sûr, identiques pour un système manuel de documentation, mais Teichroew résume les griefs qu'on peut lui faire de la façon suivante :

- Temps perdu considérable quand on change quelque chose dans la description du problème
- Mode de travail essentiellement séquentiel, négligeant les possibilités de parallélisme des activités
- Risque important d'erreurs, difficilement décelables
- Problèmes de coordination entre les personnes collaborant au projet
- Validation quasi-impossible de la solution retenue et décrite

(Teichroew, Rataj, Hershey, 1975).

Ce sont ces limitations - même si certaines sont contestables - qui justifient l'introduction de l'ordinateur dès la phase de conception d'un système, pour aider l'analyste dans l'établissement de ces exigences, de ces besoins et de l'image conceptuelle évoquée ci-dessus.

L'objectif de base d'un tel système de documentation est donc de

disposer d'une base de données

- fidèlement représentative du système en cours d'élaboration
- progressivement élaborée à partir des résultats obtenus au fur et à mesure du déroulement de la conception du système
- à la disposition des informaticiens et des utilisateurs en tant qu'élément de référence pour l'analyse, la construction et l'utilisation du système.



Quelles sont les composantes - ou ressources - dont doit disposer un pareil système de documentation (ou d'aide à la conception) pour réaliser cet objectif ?

### 13.3 COMPOSANTS

Dans la mesure où la conception d'un système d'informations passe, avant tout, par un travail de modélisation de la réalité (donc simplifiée et incomplète), il faut donc se mettre d'accord sur les concepts - et les règles pour les utiliser - que l'on va manipuler et que le système de documentation devra pouvoir enregistrer suivant les mêmes conventions. Il faut donc disposer d'un

#### 133.1 MODELE

Etablir celui-ci consiste à convenir d'une terminologie qui doit prendre en considération tous les aspects de la description d'un système d'information (à un niveau conceptuel s'entend) :

##### 1 le fonctionnement de ce dernier et ses caractéristiques

- statiques : les cellules de l'organisation et les informations qui en sortent et qui y entrent, ainsi que les processus de traitement, leurs entrées (INPUTS) et leurs sorties (OUTPUTS).
- dynamiques: la fréquence et le volume de ces mouvements d'entrée et de sortie ainsi que les événements qui déclencheront la mise en oeuvre d'un processus.

##### 2 la structuration

- des données : reprenant des renseignements sur la façon dont les données associées aux entrées/sorties sont structurées ou reliées
  - des traitements: pour indiquer comment les processus et sous-processus se hiérarchisent entre eux
- au niveau du détail souhaité

##### 3 quelques indications utiles à la **représentation** du système et notamment des renseignements sur les formats ou sur les valeurs que peuvent prendre les données qui représenteront les éléments - informations du modèle.

Pour pouvoir exprimer les "exigences" de l'utilisateur et l'image conceptuelle qui en découle à partir des concepts du modèle, il faut un

#### 133.2 LANGAGE

qui soit à la fois orienté vers l'utilisateur, c'est-à-dire le plus naturel possible, et, dans un système automatisé, suffisamment formalisé que pour être "analysé" par l'ordinateur (reconnu par un automate).

Le troisième composant d'un système d'aide à la conception est une



### 133.3 METHODE ou procédure à suivre

Celle-ci, qui propose une organisation de l'activité de conception et une approche bien définie du problème (Top-Down, Bottom up, par les données, etc.), est indispensable pour discipliner - et donc rendre plus performant - le travail des analystes. Nous pensons cependant que l'outil de documentation ou d'aide à la conception (4ème composante) doit être le plus indépendant possible de toute méthode - imposée a priori - pour garantir sa souplesse d'utilisation.

Construit uniquement à partir du modèle et de son langage, cet outil rendra ainsi les mêmes services aux analystes indépendamment de la méthode utilisée (soit propre à l'analyste, soit appropriée au type de système à étudier) avec comme seule contrainte qu'ils emploient la même terminologie (modèle) et l'expriment de la même façon (langage).

On pourrait nous objecter que l'adaptation de l'outil à une méthode donnée offre des avantages non négligeables, en particulier l'automate peut ne plus se contenter d'un rôle d'enregistrement et devenir un réel guide pour l'analyste.

Il faudrait pour cela que la méthode soit à ce point élaborée qu'elle soit d'utilisation absolument générale et adaptable à tous les cas de figure (type d'application, mentalité des analystes,...).

Or, si cet objectif <sup>est</sup> loin d'être atteint au niveau même de concepts manipulés (terminologie convenue), il nous paraît encore plus aléatoire en ce qui concerne les méthodes de travail.

La solution intermédiaire consisterait peut-être à construire, dans un premier temps, un outil indépendamment de toute méthode à partir donc d'une terminologie convenue et de son langage de manipulation et ensuite, mais ensuite seulement, d'éventuellement prévoir des "interfaces" (programmes, autres langages...) propres à chaque (ou à une) méthode employée.

Enfin, la quatrième composante est constituée d'un certain nombre d'

### 133.4 OUTILS

Ceux-ci constituent - in se - un software spécifique ou ensemble de programmes, c'est-à-dire l'instrument de mise en oeuvre et d'exploitation du système-automatisé - d'aide à la conception.

Nous avons - provisoirement - retenu 4 fonctions principales que tout outil d'aide à la conception - digne de ce nom - devrait assurer. Seule la première est vraiment indispensable pour parler de système AUTOMATISE de documentation. Les 3 autres justifient pleinement l'appellation du système "d'AIDE A LA CONCEPTION" puisqu'elles sont appelées à décharger l'analyste d'une partie de son travail (vérification, contrôle,...).

#### fonction de documentation :

qui recouvre les actions qui permettent de renseigner le concepteur et les automates sur l'état du système d'informations. Pour mettre en oeuvre cette fonction, il faut

- construire physiquement la base de données correspondant au modèle convenu
- disposer des moyens pour créer, mettre à jour et interroger cette



base de données (installation du langage et de son "compilateur").

Ceci implique que le PSA (°) puisse restituer des informations contenues dans la base (...). Le software doit permettre d'ajouter de nouvelles informations, les supprimer ou les modifier (...).

(Teichroew, Rataj, Hershey, 1975).

### **fonction d'intégration (contrôle interne) :**

qui recouvre l'ensemble des actions à mettre en oeuvre pour s'assurer de la cohérence de l'insertion d'une occurrence d'un concept (du modèle) dans l'ensemble des occurrences déjà admises pour le même concept. On peut imaginer des contrôles s'appuyant sur le respect des contraintes de cardinalité des relations et sur les désignations de ces occurrences : voir si elles sont uniques, s'il y a risque ou détection de synonymes ou de polysèmes.

Fonction d'intégration, (...) Les premiers types de contrôles correspondent à la détection des synonymes involontaires, les seconds à la détection des polysèmes.

(Rolland, 1977).

Ce software dispose également des moyens (...) pour déterminer si les noms des données ou des traitements sont uniques pour la partie de l'application définie.

(Teichroew, Rataj, Hershey, 1975).

### **fonction de contrôle (externe) de cohérence :**

qui englobe les actions permettant de s'assurer de la correction et de la fidélité de l'image descriptive dans le modèle d'un "phénomène du monde réel" tel qu'il est perçu par l'organisation. En d'autres mots, il faut donc s'assurer que l'analyste, en mettant en oeuvre les concepts du modèle retenu, donne une représentation documentaire fidèle, cohérente et conforme du phénomène réel observé dans l'organisation (séquence réelle, disponibilité des données, au moment voulu,...).

Un énoncé peut être déclaré correct s'il est cohérent et s'il est une image fidèle du phénomène réel décrit.

(Rolland, 1977).

En plus, le PSA (x) exécute un certain nombre d'analyses des relations entre les données et les processus de traitement, plus que ne pourrait en faire un analyste.

(Teichroew, Rataj, Hershey, 1975).

Le lecteur se sera rendu compte que dans le premier cas il s'agit d'un contrôle de cohérence interne du modèle alors que dans le second cas l'outil est destiné à contrôler la cohérence externe du modèle. La distinction sera peut-être mieux cernée par l'illustration suivante :

---

(°) Cfr. ISDOS dans la seconde partie de ce travail.



La fonction d'intégration a pour mission de rejeter les occurrences d'objets non conformes aux règles de manipulation - convenues - des concepts tandis que la fonction de contrôle (de la cohérence externe) se limitera à détecter (et signaler) des risques d'erreurs (de jugement de l'analyste) que le modèle trouve cependant correctes et admet.

### **fonction de contrôle d'opérationnalité :**

qui englobe les actions qui permettent de vérifier l'opérationnalité du système tel qu'il est documenté (= modèle de la réalité) par une simulation sur un "modèle réduit" par exemple.

Cette idée étant assez neuve - on ne la retrouve dans aucun des 8 systèmes que nous avons plus spécialement abordés, nous avons assez peu de renseignements sur des réalisations existantes (°) et même sur les problèmes que soulève la réalisation d'un tel outil.

Enfin, mais bien que cela ne soit guère souligné par ailleurs, nous estimons que la liaison d'un tel outil (réalisation maquette ou prototype) et d'un système de documentation est souhaitable puisque la plupart des renseignements utiles à cet outil sont disponibles.

Cette liste de fonctions n'a pas la prétention d'être exhaustive et on pourrait très bien imaginer d'autres fonctions. Par ailleurs, ces 4 fonctions elles-mêmes n'ont été, en fait, que superficiellement évoquées et devraient faire l'objet d'une étude plus fouillée qui - à notre connaissance - manque dans la littérature actuellement disponible.

## **13.4**

### **COÛTS ET BÉNÉFICES**

L'implantation et l'usage d'un système - automatisé - de documentation entraîne certains coûts qu'il faut mettre en balance avec les avantages ou bénéfices qui en découlent.

Les coûts sont principalement entraînés par

- l'utilisation de ressources informatiques (temps-machines, occupation-mémoire,...) pour exécuter les instructions du langage, maintenir une base de données et générer les "outputs" souhaités.
- un temps de formation et de rodage du personnel.
- le temps destiné à créer et maintenir un tel système ou les coûts suscités par l'achat ou la location de celui-ci.

Mais dans la mesure où toutes les possibilités d'un système de documentation sont réellement exploitées pour tous les travaux informatiques - et l'expérience prouve que ceci est loin d'être trivial, les avantages retirés par son utilisation sont appréciables. Ces avantages se traduisent notamment par

- une qualité accrue de la documentation (facilité de création et de modification...)
- une meilleure communication entre les utilisateurs et les informaticiens

---

(°) Voir à ce propos l'article de PECCOUD,  
dans Informatique et Gestion, n° 85, avril 1977.

- une réduction du temps d'analyse puisqu'une partie de celle-ci est assurée par le système. Cette réduction est normalement plus tangible en cas de modification ultérieure du système.
- une réduction également du temps de construction (programmation et organisation des données) du système dont on possède une image claire, complète et sans ambiguïté.

Le principal avantage doit normalement se trouver au niveau d'une meilleure définition ou description du problème et de la garantie d'un système d'informations mieux adaptées aux besoins de l'organisation. (°)

---

(°) Nous ne pouvons manquer de signaler la grande difficulté inhérente à l'estimation des coûts et, a fortiori, des bénéfices susceptibles d'être retirés d'un tel système.

On retrouvera dans les articles suivants (repris dans COUGER, KNAPP; Systems Analysis techniques; Wiley, 1974) quelques critères d'appréciation :

EMERY; Cost/Benefit analysis of information systems;

BOYD, KRASNOW; Economic Evaluation of MIS;

SHARPE; The cost and effectiveness of computer systems;

GREGORY, VAN HORN; Value and Cost of Information;

FRIED; Estimating the cost of system implementation;



## SYNTHESES de SYSTEMES EXISTANTS.

---

Cette partie n'a pas la prétention de dresser un synopsis complet des systèmes de documentation que nous avons rencontrés, elle se limitera à la présentation de 8 systèmes d'aide à la conception que nous avons plus spécialement abordés. Il s'agit de ADS, TAG, MINOS, REMORA, DATA DICTIONARY, CASCADE, CADIS et ISDOS.

De plus, nous ne retiendrons de ces systèmes que ce qu'ils proposent pour aider l'analyste dans son travail de CONCEPTION, nous refusant d'envisager tout ce qui, éventuellement dans ces mêmes systèmes, touche aux phases de REALISATION et d'EXPLOITATION opérationnelle.

D'autre part, nous aurions pu reprendre, par système, un simple résumé des articles dont nous disposions; nous avons cependant préféré prendre comme grille d'analyse unique de ces 8 systèmes, la découpe que nous venons d'établir dans la première partie (Spécifications d'un système de documentation).

Aussi aborderons-nous successivement pour chacun de ces systèmes:

- les concepts qu'il propose pour expliquer (modéliser) le fonctionnement du système, la structuration des données et des traitements ainsi que les quelques éléments utiles à la représentation des éléments.
- le langage pour manipuler ces concepts (pour modéliser la réalité)
- quelques indications sur l'éventuelle méthode, jointe au système, mais exposées très sommairement vu la remarque que nous avons faite sur l'indépendance d'un outil de documentation par rapport à une quelconque méthode,
- et les outils proposés, dans le contexte d'une plus ou moins grande automatisation, pour assurer les fonctions de documentation, d'intégration et de contrôle de cohérence.

Enfin, nous ne pouvions terminer sans signaler - même si cela sort du cadre que nous nous sommes fixés - quelques prolongements possibles, existants ou en voie de réalisation, à partir d'un système de documentation :

- soit vers une automatisation complète du processus d'élaboration d'un système automatisé d'informations (conception et réalisation) : ISDOS.
- soit vers une aide accrue au gestionnaire ou à l'organisateur par la mise en évidence, à partir des "flux d'informations", de la STRUCTURE de CONTROLE de l'organisation (Bourgeois, 1977). (°)

---

(°) On pourrait s'étonner du choix d'un tel système inspiré des travaux de Forrester - moins connu que les autres systèmes retenus. En fait, ce système a été sélectionné à titre exemplatif d'une voie originale - peut-être pas le plus représentatif mais le seul dont nous disposions.

Nous voulions souligner par là que cette voie -vers une intégration du système d'informations dans l'organisation- gagnerait certainement à être exploitée.



## 2.1

**ADS.****ACCURATELY DEFINED SYSTEM.**

Quoiqu'à l'origine, l'intervention de l'ordinateur n'ait pas été envisagée, ce système, initialement conçu par NCR (1968), a depuis été repris par d'autres institutions et a fait l'objet d'une plus ou moins grande automatisation.

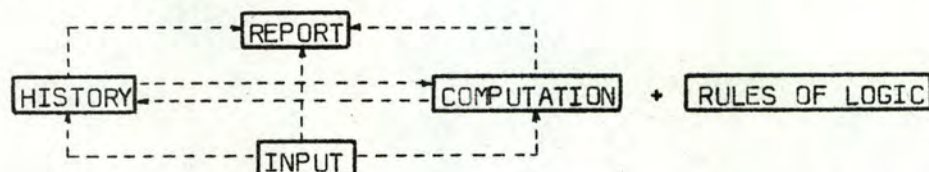
ADS a une vocation essentiellement documentaire et a pour principale préoccupation de faciliter - et de discipliner - par un enregistrement systématique la DEFINITION et la COMMUNICATION des exigences FONCTIONNELLES des utilisateurs à propos du système d'informations (ou de l'application) en construction.

**MODELE**

Ce modèle simple et sans prétention théorique s'attache essentiellement à décrire le fonctionnement - statique - du système.

## ° Fonctionnement

Le schéma suivant met en évidence les principaux éléments de ce modèle et leurs relations



Chacun de ces éléments est décrit à partir d'un certain nombre de DATA (ou variables) repérées par un nom unique dans tout le système.

Les relations entre ces différents éléments sont assurées par un CHAINAGE qui relie toutes les utilisations d'une DATA au sein d'un élément à sa SOURCE, c'est-à-dire une utilisation logiquement précédente de la même DATA dans la définition d'un autre élément. Ainsi, en fin de description, toutes les utilisations d'une DATA sont chaînées d'un REPORT à un INPUT, ce qui facilite le repérage d'omissions ou de contradictions dans le système tel qu'il est défini.

Les REPORTS sont des états imprimés produits par la système, à l'exclusion d'autres formes possibles de sorties.

Les INPUTS regroupent les données à saisir - nécessaire à la production des résultats; le modèle autorise également l'expression de VALIDATION RULES, règles de validité auxquelles certaines DATA de ces INPUTS doivent satisfaire.

Par HISTORY, il faut comprendre un ensemble de données mémorisées (notion classique de fichier) qui traitées à un moment donné doivent être conservées en vue d'un traitement ultérieur.

COMPUTATION prend en considération l'aspect de traitement des DATA, y compris les CONDITIONS sous lesquelles chaque RULES est exécutée.

L'élément RULES OF LOGIC est introduit dans le modèle pour faciliter l'expression de règles logiques plus complexes de calcul ou même de production d'états.



D'autre part, alors que l'accent semble avoir été mis sur le fonctionnement du système, l'aspect dynamique semble avoir été sinon négligé, du moins très peu développé. Les possibilités de description se limitent à

- quelques indications de volume
- l'expression des conditions sous lesquelles un état est produit
- et l'indication d'une période de rétention pour les DATA d'un HISTORY, c'est-à-dire pour les données mémorisées.

#### ° Structuration

Ce modèle permet une description complète - quoiqu'un peu dépassée à l'heure actuelle - de la structure d'un état imprimé (REPORT) mais, par contre, n'autorise qu'une très pauvre structuration des données mémorisées (HISTORY) et des traitements (COMPUTATION).

#### ° Représentation

Au fil des utilisations d'une DATA, celle-ci sera complétée par

- des indications de format (de saisie et d'édition)
- et éventuellement par l'ensemble des valeurs qu'elle peut prendre (MEMO LISTE au niveau des INPUTS).

### METHODE

Nous ne pouvons manquer de signaler qu'ADS a également choisi comme prémices à sa mise en oeuvre une approche "par les sorties". Le point de départ de la description d'un système d'informations est donc l'identification des REPORTS ou états imprimés à produire; et, à partir de là, il reste à déterminer les INPUTS, COMPUTATIONS et autres HISTORIES.

### LANGAGE

A chacun des (5) éléments principaux correspond un descripteur sous forme tabulaire.

Le descripteur des REPORTS est donc destiné à enregistrer la structure et le contenu d'un état imprimé d'une façon précise avec ses sections

de description des lignes de l'état (titre, rubriques et intitulés)  
de dynamique d'édition et de niveau de rupture  
de définition des conditions d'édition de l'état.

Le descripteur des INPUTS

identifie l'entrée

décrit celle-ci sous un format "image-carte"

définit plus précisément les rubriques ou DATA

et donne les différentes valeurs possibles d'une donnée.

La présentation du descripteur des HISTORIES ne diffère guère du descripteur précédent.

Tandis que le descripteur des COMPUTATIONS permet de prendre en considération les opérations algébriques sur les données de la forme

si <condition> alors DATA 1 = DATA 2 <opérateur> DATA 3.

Le descripteur RULES OF LOGIC n'est, en fait, destiné qu'à enregistrer des tables de décisions sous une forme matricielle.



Et puisque l'objectif est d'établir un lien entre les DATA en partant du REPORTS pour aboutir aux INPUTS, toute mention d'une DATA au sein d'un descripteur doit automatiquement être accompagnée de l'indication de sa provenance.

Et, toute utilisation (et donc provenance) d'une DATA est identifiée par

- le nom du type de descripteur (Report, Input, Computation, History)
- le numéro de page pour ce type de descripteur
- et le numéro de ligne dans cette page où la DATA est utilisée.

**INPUT DEFINITION for** Application: NAME OF INPUT: RECORD: PREPARED BY: DATE: PAGE: OF:

**FOR FURTHER FORMATS ENTER 10 CODE**

**NAME OF FORMAT: EMPLOYEE TIME REPORT**

DATE	PROJECT NUMBER	CLOCK NUMBER	EMPLOYEE NAME

**III CROSS REFERENCE TO FIELD NAMES**  
(WHEN FIELD HEADING INCLUDES MULTIPLE CODES LIST ALL CODES ON ATTACHMENT)

NAME	SOURCE	VALUE
GROSS WAGES		

**I REPORT LAYOUT**

**COMPUTATION DEFINITION**

NAME OF PROCEDURE: PREPARED BY: DATE: PAGE: 01 OF 04

**IF CODE (N,01,07) = 1 IF NOT GO TO C,01,04**

**IF CODE (N,01,07) = 2 IF NOT GO TO C,01,07**

**GROSS WAGES**

**GROSS WAGES**

Exemple d'utilisation d'un descripteur ADS.

## OUTIL

Tant que le système était manuel la fonction de documentation (collecte, mémorisation, interrogation) était assurée par 5 documents représentant les DESCRIPTEURS du langage.

Dans une version automatisée les descripteurs sont perforés et l'ordinateur intervient pour saisir ces données, exercer certains contrôles de cohérence et générer une documentation claire et complète à propos du système en construction.

Le texte-source est donc analysé par programme lequel

- s'assure de la validité syntaxique des lignes ADS et opère la détection de certaines erreurs
- construit un dictionnaire des données et de leurs utilisations
- exerce une fonction de contrôle de cohérence pour découvrir certaines erreurs ou risques d'erreur dans le chaînage des DATA
- génère des matrices
  - d'antériorité d'une DATA par rapport à une autre
  - d'utilisation d'une DATA par un processus (computation)
- produit une série de rapport
  - dictionnaire des données
  - matrices d'antériorité et d'utilisation



- diagnostics et messages d'erreurs
  - et les 5 types de descripteurs clairs et cohérents dans une forme très proche des documents saisis.
- Ces listings de documentation constituent alors le stock documentaire pouvant être compulsé (interrogé) par les utilisateurs ou analystes.

Signalons que R. Lesuisse a proposé quelques modifications mineures pour remettre ADS aux goûts du jour; pour notamment permettre de mieux structurer les états imprimés (arborescence : état, groupes, lignes, rubriques) et les traitements (organisation arborescente des opérations).

Enfin, la lecture de (Nunamaker, 1976) est intéressante dans la mesure où elle montre une utilisation de la méthode ADS dans le contexte d'un projet réel et donne quelques indications sur les avantages qu'on est en droit d'attendre de l'implantation d'un tel système de documentation.



**TAG.****TIME AUTOMATED GRID (IBM)**

TAG est un outil automatisé qui tente d'aider l'analyste dans la collecte des données, l'analyse des exigences en donnée et la définition des "flux" de données.

La principale particularité de TAG est de pouvoir dégager, sur base des moments physiques (temps) de saisie d'INPUT ou de génération d'OUTPUT, les données devant être mémorisées (FILE).

**MODELE**

Ce modèle, au moins aussi simple que celui d'ADS, permet juste de décrire le fonctionnement du système à partir des documents saisis ou produits dans le système.

## ° Fonctionnement

Ce modèle ne repose que sur 3 concepts :

- OUTPUT ou résultat à produire
- INPUT ou entrée nécessaire pour la production des résultats
- FILE ou données mémorisées

et ce qui frappe peut-être le plus, c'est l'absence de concept permettant de prendre en considération les traitements ou calculs.

Les 3 concepts, représentatifs de documents saisis ou générés, sont décrits à partir d'un certain nombre de DATA (ou variables) caractérisées notamment par:

- un nom
- une classe d'utilisation signalant que cette DATA intervient ou non dans un calcul, comme résultat ou facteur de ce calcul.

Mais ce qui fait l'originalité de TAG, c'est sa façon particulière d'intégrer la "composante dynamique" du fonctionnement.

Un élément INPUT ou OUTPUT est effectivement caractérisé par

- un triplet (fréquence, période, priorité) à partir duquel TAG calcule un "numéro de cycle" qui fixe le moment physique de saisie ou de génération du document correspondant dans le système.
- Une indication de volume, associée à la fréquence, nécessaire plus tard (REALISATION) en vue d'optimiser les performances.

Et, en fait, pour TAG, c'est précisément un intervalle de temps, entre les moments de saisie de DATA et de génération ou sortie de ces DATA, qui justifiera la création d'un FILE ou mémorisation de données.

## ° Structuration

A ce niveau, le modèle est très pauvre

- puisqu'il ne prévoit rien en ce qui concerne les traitements
- et qu'il ne permet guère que de signaler qu'une DATA est éventuellement utilisée pour contrôler la séquence d'un INPUT ou d'un OUTPUT.

## ° Représentation

A ce stade, seules quelques indications de taille et de format viennent compléter les caractéristiques d'une DATA.



## LANGUAGE

Un seul type de descripteur, sous forme tabulaire, permet de décrire le système au moyen des concepts du modèle.

La première partie de ce descripteur permet d'enregistrer les renseignements concernant l'élément INPUT, OUTPUT ou FILE :

- son nom et son appartenance à une classe (I-O-F)
- son triplet (fréquence, période, priorité)
- et des indications de volumes.

Tandis que la seconde partie concerne les DATA qui composent l'élément (INPUT, OUTPUT ou FILE) et permet de mentionner par variable :

- son nom
- sa taille ou son format
- sa classe d'utilisation
- et quelques autres indications.

IBM		TIME AUTOMATED GRID TECHNIQUE (TAG)		INPUT/OUTPUT ANALYSIS FORM		PAGE _____ OF _____			
<div>DATA TYPE</div> <div>CODE</div> <div>FREQUENCY</div> <div>REASON</div> <div>PRIORITY</div> <div>PROGRAM</div> <div>ALLOCATION</div>		<div>REQUIREMENT TITLE</div>		<div>PEAK (P)</div> <div>VOLUME</div> <div>FREQUENCY</div> <div>PERIOD</div>		<div>AVERAGE (A)</div> <div>VOLUME</div> <div>FREQUENCY</div> <div>PERIOD</div>			
<div>DATE</div>		<div>PREPARED BY</div>		<div>MONTH, YEAR, DAY</div> <div>VOLUME</div> <div>FREQUENCY</div> <div>PERIOD</div>		<div>MONTH, YEAR, DAY</div> <div>VOLUME</div> <div>FREQUENCY</div> <div>PERIOD</div>			
<div>COMMENT</div>		<div>DATA NAME</div>		<div>SIZE</div> <div>FILE TYPE</div> <div>RATIO</div> <div>SEQUENCE</div> <div>FORMAT</div>		<div>REMARKS</div>			
<div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> <div>13</div> <div>14</div> <div>15</div> <div>16</div> <div>17</div> <div>18</div> <div>19</div> <div>20</div> <div>21</div> <div>22</div> <div>23</div> <div>24</div> <div>25</div> <div>26</div> <div>27</div> <div>28</div> <div>29</div> <div>30</div> <div>31</div> <div>32</div> <div>33</div> <div>34</div> <div>35</div> <div>36</div> <div>37</div> <div>38</div> <div>39</div> <div>40</div> <div>41</div> <div>42</div> <div>43</div> <div>44</div> <div>45</div> <div>46</div> <div>47</div> <div>48</div> <div>49</div> <div>50</div> <div>51</div> <div>52</div> <div>53</div> <div>54</div> <div>55</div> <div>56</div> <div>57</div> <div>58</div> <div>59</div> <div>60</div> <div>61</div> <div>62</div> <div>63</div> <div>64</div> <div>65</div> <div>66</div> <div>67</div> <div>68</div> <div>69</div> <div>70</div> <div>71</div> <div>72</div> <div>73</div> <div>74</div> <div>75</div> <div>76</div> <div>77</div> <div>78</div> <div>79</div> <div>80</div> <div>81</div> <div>82</div> <div>83</div> <div>84</div> <div>85</div> <div>86</div> <div>87</div> <div>88</div> <div>89</div> <div>90</div> <div>91</div> <div>92</div> <div>93</div> <div>94</div> <div>95</div> <div>96</div> <div>97</div> <div>98</div> <div>99</div> <div>100</div>		<div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> <div>13</div> <div>14</div> <div>15</div> <div>16</div> <div>17</div> <div>18</div> <div>19</div> <div>20</div> <div>21</div> <div>22</div> <div>23</div> <div>24</div> <div>25</div> <div>26</div> <div>27</div> <div>28</div> <div>29</div> <div>30</div> <div>31</div> <div>32</div> <div>33</div> <div>34</div> <div>35</div> <div>36</div> <div>37</div> <div>38</div> <div>39</div> <div>40</div> <div>41</div> <div>42</div> <div>43</div> <div>44</div> <div>45</div> <div>46</div> <div>47</div> <div>48</div> <div>49</div> <div>50</div> <div>51</div> <div>52</div> <div>53</div> <div>54</div> <div>55</div> <div>56</div> <div>57</div> <div>58</div> <div>59</div> <div>60</div> <div>61</div> <div>62</div> <div>63</div> <div>64</div> <div>65</div> <div>66</div> <div>67</div> <div>68</div> <div>69</div> <div>70</div> <div>71</div> <div>72</div> <div>73</div> <div>74</div> <div>75</div> <div>76</div> <div>77</div> <div>78</div> <div>79</div> <div>80</div> <div>81</div> <div>82</div> <div>83</div> <div>84</div> <div>85</div> <div>86</div> <div>87</div> <div>88</div> <div>89</div> <div>90</div> <div>91</div> <div>92</div> <div>93</div> <div>94</div> <div>95</div> <div>96</div> <div>97</div> <div>98</div> <div>99</div> <div>100</div>		<div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> <div>13</div> <div>14</div> <div>15</div> <div>16</div> <div>17</div> <div>18</div> <div>19</div> <div>20</div> <div>21</div> <div>22</div> <div>23</div> <div>24</div> <div>25</div> <div>26</div> <div>27</div> <div>28</div> <div>29</div> <div>30</div> <div>31</div> <div>32</div> <div>33</div> <div>34</div> <div>35</div> <div>36</div> <div>37</div> <div>38</div> <div>39</div> <div>40</div> <div>41</div> <div>42</div> <div>43</div> <div>44</div> <div>45</div> <div>46</div> <div>47</div> <div>48</div> <div>49</div> <div>50</div> <div>51</div> <div>52</div> <div>53</div> <div>54</div> <div>55</div> <div>56</div> <div>57</div> <div>58</div> <div>59</div> <div>60</div> <div>61</div> <div>62</div> <div>63</div> <div>64</div> <div>65</div> <div>66</div> <div>67</div> <div>68</div> <div>69</div> <div>70</div> <div>71</div> <div>72</div> <div>73</div> <div>74</div> <div>75</div> <div>76</div> <div>77</div> <div>78</div> <div>79</div> <div>80</div> <div>81</div> <div>82</div> <div>83</div> <div>84</div> <div>85</div> <div>86</div> <div>87</div> <div>88</div> <div>89</div> <div>90</div> <div>91</div> <div>92</div> <div>93</div> <div>94</div> <div>95</div> <div>96</div> <div>97</div> <div>98</div> <div>99</div> <div>100</div>		<div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> <div>13</div> <div>14</div> <div>15</div> <div>16</div> <div>17</div> <div>18</div> <div>19</div> <div>20</div> <div>21</div> <div>22</div> <div>23</div> <div>24</div> <div>25</div> <div>26</div> <div>27</div> <div>28</div> <div>29</div> <div>30</div> <div>31</div> <div>32</div> <div>33</div> <div>34</div> <div>35</div> <div>36</div> <div>37</div> <div>38</div> <div>39</div> <div>40</div> <div>41</div> <div>42</div> <div>43</div> <div>44</div> <div>45</div> <div>46</div> <div>47</div> <div>48</div> <div>49</div> <div>50</div> <div>51</div> <div>52</div> <div>53</div> <div>54</div> <div>55</div> <div>56</div> <div>57</div> <div>58</div> <div>59</div> <div>60</div> <div>61</div> <div>62</div> <div>63</div> <div>64</div> <div>65</div> <div>66</div> <div>67</div> <div>68</div> <div>69</div> <div>70</div> <div>71</div> <div>72</div> <div>73</div> <div>74</div> <div>75</div> <div>76</div> <div>77</div> <div>78</div> <div>79</div> <div>80</div> <div>81</div> <div>82</div> <div>83</div> <div>84</div> <div>85</div> <div>86</div> <div>87</div> <div>88</div> <div>89</div> <div>90</div> <div>91</div> <div>92</div> <div>93</div> <div>94</div> <div>95</div> <div>96</div> <div>97</div> <div>98</div> <div>99</div> <div>100</div>		<div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> <div>13</div> <div>14</div> <div>15</div> <div>16</div> <div>17</div> <div>18</div> <div>19</div> <div>20</div> <div>21</div> <div>22</div> <div>23</div> <div>24</div> <div>25</div> <div>26</div> <div>27</div> <div>28</div> <div>29</div> <div>30</div> <div>31</div> <div>32</div> <div>33</div> <div>34</div> <div>35</div> <div>36</div> <div>37</div> <div>38</div> <div>39</div> <div>40</div> <div>41</div> <div>42</div> <div>43</div> <div>44</div> <div>45</div> <div>46</div>	

Modèle d'un descripteur d'entrée/sortie.

## METHODE

Tout comme ADS, TAG suppose qu'on définisse tous les OUTPUTS avant de fixer les INPUTS nécessaires et éventuellement les FILES. C'est avec l'aide des rapports générés par TAG que l'analyste déterminera systématiquement les INPUTS, et, ultérieurement sur base des intervalles de temps, les FILES nécessaires.

## OUTIL

Les fonctions de documentation et de contrôle de cohérence sont partiellement prises en considération par TAG qui, à partir des cartes perforées représentant les descripteurs, génère 10 rapports, résultats de l'analyse qu'il fait subir aux descripteurs.



- Rapport 1 (User's data) : reprend tels quels les descripteurs-sources
- Rapport 2 (Glossary) : est un dictionnaire de données (DATA)
- Rapport 3 (Document) : est une liste des éléments (INPUT-OUTPUT) du système et de leurs caractéristiques, classés selon le triplet (fréquence, période, priorité).
- Rapport 4 (list of data names) : reprend par DATA toutes les utilisations qui en sont faite au sein d'éléments (I/O).
- Rapport 5 (Time-Grid analyses) : place toutes les survenances d'une DATA sur une échelle-temps et indique les DATA devant être mémorisées.
- Rapport 6 (Unresolved conditions) : constitue un diagnostic des erreurs probables de définition par l'analyste.
- Rapport 7 (Time Key analysis) : reprend les titres d'élément (I/O) sur une échelle-temps, avec leurs caractéristiques de volume notamment.
- Rapports 8 et 9 (Fils Records) : constituent des propositions de Fichiers (FILE) avec tentatives d'optimisation.
- Rapport 10 (JOB DEFINITION) : regroupe tous les INPUTS, OUTPUTS et FILES par intervalle de temps.

TAG n'a pas la prétention d'automatiser le processus de conception d'un système d'informations, ces rapports permettent tout au plus d'aider l'analyste dans son travail de maintenance des "exigences en informations" - cohérentes - des utilisateurs.

L'avantage du contrôle dynamique des mouvements de données est certain mais reste limité (à ce qui est périodique) et ne doit pas masquer la pauvreté du modèle, dont nous avons fait état dans le paragraphe concerné.



## SYSTEME GENERALISE DE DOCUMENTATION.

Précisons, peut-être, qu'il s'agit d'un système que nous connaissons un peu mieux que d'autres (systèmes) pour l'avoir pratiqué pendant les 4 mois de notre stage chez Peugeot (Thuly). On comprendra dès lors que nous avons réduit sa présentation au minimum, des commentaires à son sujet ayant déjà fait l'objet d'un rapport.

- une indication du type d'élément (Mot, lot...)
- une racine ou identifiant unique
- une définition réduite qui permet une identification claire et complète
- une description complète sous forme de commentaire
- un code auteur reprenant l'analyste responsable de l'élément défini.

Enfin, un code nature permet d'identifier chaque caractéristique relevée au niveau d'un de ces éléments.

STRUCTURE LOGIQUE

LOT

MAT

Un LOT correspond à l'association de mots ayants par elle-même un contenu sémantique qui ne dépend que de la forme de la combinaison de mots et à laquelle on donne une désignation qui permet de la



distinguer d'autres associations de mots, combinés différemment. Signalons qu'il a été introduit, plus récemment, une classification des LOTS en 4 catégories, en fonction du rôle qu'ils jouent dans la description des objets à décrire :

PERMANENT, SITUATIONNEL, EVENEMENTIEL, HISTORIQUE.

D'autre part, partant de la constatation, faite également par CASCADE, que plusieurs "messages" ou "ensembles d'informations" peuvent avoir la même structure logique, Minos propose de décrire une seule fois cette structure (la racine) et d'en préciser les différents NIVEAUX D'ELABORATION possibles (même racine + indication numérique de ce niveau).

Ce modèle permet également une hiérarchisation des traitements (PROCEDURE), du plus général au plus détaillé. Mais si Minos-SGDG admet un nombre indéfini de niveaux, la méthode Minos en retient principalement 3: (gamme, séquence, règle) ou plus récemment : (application, fonction logique et processus); (Thuly, Saunier, 1976).

#### ° Fonctionnement

Ce modèle offre la possibilité à l'analyste de déclarer qu'une structure logique, un lot ou un mot intervient dans une procédure à titre d'entrée, de sortie ou de fichier intermédiaire (suffixe d'utilisation).

Signalons, d'autre part, l'existence d'un élément dont nous n'avons pas parlé jusqu'à présent et qui est celui de DIAGRAMME de PARTICIPATION qui est une représentation de la logique fonctionnelle et des relations qui expriment le fonctionnement du système (antériorité des données les unes par rapport aux autres, utilisation des données par les traitements...).

Enfin, le caractère dynamique se limite, comme la plupart des systèmes, à la prise en considération

- de la fréquence et de la période de génération des structures logiques ou d'activation des procédures
- des volumes de réalisation lors d'une génération.

#### ° Représentation

Contentons nous de signaler que le modèle contient les composants nécessaires pour compléter la description d'un Mot : format, ensemble de valeurs possibles, synonymes...

### LANGAGE

A chacun des 5 éléments principaux du modèle correspond un descripteur sous forme tabulaire qui permet de décrire tous les renseignements concernant ces éléments.

L'unité de base de description est, en fait, la "ligne" de descripteur puisqu'on reprend au niveau de chaque ligne

- une indication du type d'élément (mot, lot, structure, procédure, diagramme), couplée à une indication du type de traitement (CREATION, SUPPRESSION, MISE A JOUR d'un élément ou d'une ligne).
- la racine
- le code auteur

et c'est le code nature (ou numéro de ligne) qui indique le type de renseignement (la caractéristique) contenu dans la ligne au sein



Mais le simple examen du document ci-joint permettra de mieux visualiser la présentation ci-dessous.

Récapitulons les principaux renseignements que l'on peut trouver sur les descripteurs, compte tenu de notre intérêt pour la seule phase de CONCEPTION et sans reprendre les caractéristiques communes à tous les éléments (racine, code auteur, définition réduite et complète, type de descripteur...):

- descripteur des mots : format (dimension et mode de représentation)  
contrôle sur l'information  
valeurs déterminantes (code et explication)
- descripteur des lots : contrôle sur l'information  
composants de la structure (soit des lots,  
soit des mots)
- descripteur des structures : périodicité et fréquence, volumes  
niveaux d'élaboration (les "utilisations"  
d'une structure dans un message)  
composants de la structure (des lots)  
liste des bénéficiaires et responsables
- descripteur des procédures : fréquence d'activation  
modules ou procédures appelées (hiérarchisation des traitements)  
structures manipulées (racines + suffixe  
d'utilisation : in, out, intermédiaires)  
Mots ou lots utilisés (in, out ou interval).
- descripteurs des diagrammes : les résultats à atteindre  
les procédures pour y arriver  
les informations (entrée/interne) nécessaires  
les règles de conjugaison des informations.

## OUTIL

Au niveau qui nous retient (conception), seule la fonction de documentation est assurée tandis que la fonction d'intégration ne l'est que partiellement puisque jusqu'à présent - mais c'est sur les tablettes de travail de l'équipe "méthodes" de J. THULY - aucune détection des polysèmes et des synonymes n'est possible.

Le modèle est donc physiquement représenté par une organisation de fichiers classiques (5 répertoires).

Les descripteurs du langage sont perforés et contrôlés, permettent de mettre à jour les fichiers, et génèrent éventuellement des messages d'erreurs.

Une série d'instructions (en fait, des noms de procédures de JCL) donnent à l'utilisateur un très grand choix de "rapports" et "d'interrogations" possibles. Les principaux types d'états sont la

- fiche signalétique reprenant tous les renseignements à propos d'un élément ou le
- glossaire identifiant les éléments d'un même type (même répertoire).

Ces publications se font suivant certains critères de sélection ou certains critères de dépendance des informations les unes par rapport aux autres.

Il faut cependant savoir que de nombreux autres outputs et donc des programmes (outils) sont disponibles dans le SGDG et rendent de grands services lors des autres phases d'élaboration d'un système d'informations, en particulier lors de l'"exploitation" - courante et opérationnelle - du système complètement construit.



1 2  
3=C  
C=M  
L=5

		10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78							
IDENTIFICATION	CODE NATURE																																																																												
EXPLOITATION CLASSEMENT DISTRIBUTION	NIV. PERMIT N. FEUILLES N. LIGNE N. EX. ELEMENTS DE CLASSEMENT R. RACINE DE CLASS <sup>MS</sup> DESTINATAIRE																																																																												
C.NAT.: 050 à 099																																																																													
NIVEAUX D'ELABORATION																																																																													
C.NAT.: 100 à 199																																																																													
ORGANISATION PHYSIQUE	BLKSIZE LRECL RECFM ALLOC. PRIM. ALLOC. SECOND. DATE DE CREATION DATE DE DISPONIB. DATE D'EXPIRATION VOLUME PHYSIQUE VOLUME LOGIQUE NBR. SIT																																																																												
C.NAT.: 200 à 229																																																																													
VOLUMES DSNAMES	NUMERO DE VOLUME LABEL FICHIER (DSNAME)																																																																												
C.NAT.: 230 à 249																																																																													
SAUVEGARDES ET RESPONSABLES																																																																													
C.NAT.: 250 à 299																																																																													
ENREGISTREMENTS	R. RACINE R. RACINE R. RACINE R. RACINE R. RACINE R. RACINE R. RACINE R. RACINE																																																																												
C.NAT.: 300 à 399																																																																													
STRUCTURES MARK 4	R. RACINE SEG. NO. R. RACINE CLE 1 R. RACINE CLE 2 R. RACINE CLE 3 SEG. OCEURS ZONE DE COMPTEURS																																																																												
C.NAT.: 400 à 499																																																																													



## DATA DICTIONARY.

DB/DC - IMS (IBM).

---

Ce DATA DICTIONARY occupe une place un peu particulière parmi les différents systèmes de documentation que nous avons sélectionnés. Fortement lié à un software de gestion de Bases de données, ce système est plus apte à rendre des services lors des phases de REALISATION que lors de la CONCEPTION du système d'informations.

Aussi est-ce moins pour son modèle que pour son langage que nous avons jugé utile de brièvement présenter ce système, épuré des composantes trop directement liées à IMS/DL 1.

### MODELE

Si ce modèle permet d'envisager une structuration - physique - assez fine des données et des traitements, il semble peu approprié pour se donner une bonne image du fonctionnement du système d'informations.

#### ° Structuration

Ainsi, la structuration des données est-elle assurée par 3 concepts organisés hiérarchiquement

- DATA BASE DL 1 ou ensembles d'informations non DL 1.
- SEGMENT DL 1 ou enregistrement d'un ensemble d'informations
- ELEMENT ou FIELD DL 1 ou rubrique élémentaire quelconque.

La structuration des traitements, quant à elle, s'organise autour des concepts de

- SYSTEM ou application
- JOB
- PROGRAM
- MODUL
- TRANSACTION

plus familiers aux analystes organiques qu'aux "concepteurs".

#### ° Fonctionnement

A ce niveau, les possibilités offertes par le modèle se limitent à pouvoir spécifier les relations d'utilisation des "données" (Data base, segment, élément) par les traitements (systems,...transaction). Il n'est donc apparemment pas possible de décrire le mode d'intervention - input, output, intermédiaire - de ces données mémorisées. De plus, la description des documents saisis ou générés par le système n'est théoriquement pas envisagée.

#### ° Représentation

Comme nous l'avons précédemment signalé, la plupart des éléments de ce modèle ont trait exclusivement à la REPRESENTATION physique du système. Ceci explique notre intérêt très limité pour les concepts de ce modèle.



Notons , enfin, que la "définition" d'un élément de ce modèle peut contenir les caractéristiques suivantes :

- le nom de l'élément et d'éventuels synonymes
- un certain nombre d'attributs (préétablis ou fixés par l'analyste)
- une description complète de l'élément ou un commentaire
- les UTILISATEURS des données
- et les relations avec les autres éléments du système.

## LANGAGE

Grâce à des commandes, semblables à des instructions de manipulation de Bases de données, l'analyste peut mettre à jour son stock documentaire.

Une commande complète comprend, outre le nom de la commande (mot réservé), les informations qui complètent ou qualifient la description de l'opération souhaitée comme, par exemple, le type et le nom de ou des éléments concernés, éventuellement complété par certains attributs de leur "définition".

Ces commandes laissent donc la possibilité à l'analyste de :

- ajouter un élément et sa "définition" (ADD)
- ajouter une relation et les éléments rattachés (ADD-RELATIONSHIP)
- changer une "définition" d'un élément (CHANGE-IN)
- changer le nom d'un élément (CHANGE-NAME)
- changer une relation et les éléments concernés (CHANGE-RELATIONSHIP-DATA)
- supprimer un élément; son nom et sa "définition" (DELETE)
- supprimer une partie de la "définition" d'un élément (DELETE-DATA)
- supprimer une relation entre 2 éléments (DELETE-RELATIONSHIP).
- supprimer une relation et les éléments concernés par cette relation (DELETE-RELATIONSHIP - DATA).
- ou supprimer une structure complète - un élément et sa descendance hiérarchique (DELETE-STRUCTURE).

On peut donc ajouter, modifier ou supprimer

- un objet (élément DATA BASE, SEGMENT,...TRANSACTION)
- une propriété d'objet (composantes d'une "définition")
- une relation entre objets (entre éléments).

## OUTIL

La fonction de documentation est complètement assumée :

- puisque le modèle est physiquement construit sous forme d'une base de données (5 DATA BASES physiques) gérée par IMS
- que le langage décrit constitue le moyen pour mettre à jour cette base.
- que d'autres commandes du langage (SCAN et REPORT) permettent d'interroger la base de données.



Signalons enfin que la gestion de ce stock documentaire peut se faire soit en batch, soit en interactif et que, d'autre part, il existe d'autres commandes (et donc d'autres outils les supportant) qui permettent notamment des transferts d'informations entre ce data dictionary et DL 1 (PSB, DBD...) et des générations de déclarations de structures de données en langage de programmation à partir du dictionnaire.

L'inadaptation du modèle et l'absence d'outils de contrôle d'intégration et de cohérence n'en font pas un système de documentation modèle, dans le créneau que nous nous sommes fixés, mais son langage offre une plus grande souplesse d'utilisation que les langages aux descripteurs de forme tabulaire. Notons encore que cette souplesse peut encore être améliorée (Cfr CASCADE ou CADIS).

```

INPUT RECORD /SETSTAT=T SETLANG=A SETOCR=O SETDBTP=P;
DBD0026 I OK**

INPUT RECORD /ADD SEGMENT EMPMAST DESC=(1,'EMPLOYEE MASTER SEGMENT')
INPUT RECORD /  BYTES=141 NUMEL=3;
DBD1002 I SEG TA EMPMAST 0 HAS BEEN ADDED
DBD1801 I DATA HAS BEEN ADDED TO SEG TA EMPMAST 0

INPUT RECORD /ADD RELATIONSHIP DATABASE PAYDB WITH SEGMENT EMPMAST+
INPUT RECORD /  FREQ=1000 PTR=TB TYPE=P RULES=(VVV);
DBD0015 I DBS TP PAYDB 0 IS SUBJECT A
DBD0015 I SEG TA EMPMAST 0 IS SUBJECT B
DBD0019 I RELATIONSHIP ESTABLISHED UNDER PRIMARY NAMES
DBD1701 I DATA HAS BEEN ADDED TO RELATIONSHIP

INPUT RECORD /ADD RELATIONSHIP SEGMENT EMPMAST WITH ELEMENT EMPNO+
INPUT RECORD /  START=1 SEQ=U;
DBD0015 I SEG TA EMPMAST 0 IS SUBJECT A
DBD0015 I DTE TA EMPNO 0 IS SUBJECT B
DBD0019 I RELATIONSHIP ESTABLISHED UNDER PRIMARY NAMES
DBD1701 I DATA HAS BEEN ADDED TO RELATIONSHIP

INPUT RECORD /ADD ELEMENT EMPNAME TYPE=C BYTES=60;
DBD1002 I DTE TA EMPNAME 0 HAS BEEN ADDED
DBD1801 I DATA HAS BEEN ADDED TO DTE TA EMPNAME 0

INPUT RECORD /ADD RELATIONSHIP SEGMENT EMPMAST WITH ELEMENT EMPNAME+
INPUT RECORD /  START=7 SEQ=G;
DBD0015 I SEG TA EMPMAST 0 IS SUBJECT A
DBD0015 I DTE TA EMPNAME 0 IS SUBJECT B
DBD0019 I RELATIONSHIP ESTABLISHED UNDER PRIMARY NAMES
DBD1701 I DATA HAS BEEN ADDED TO RELATIONSHIP

INPUT RECORD /ADD ELEMENT ADDR TYPE=C BYTES=75;
DBD1002 I DTE TA ADDR 0 HAS BEEN ADDED
DBD1801 I DATA HAS BEEN ADDED TO DTE TA ADDR 0

INPUT RECORD /ADD RELATIONSHIP SEGMENT EMPMAST WITH ELEMENT ADDR+
INPUT RECORD /  START=67 SEQ=G;
DBD0015 I SEG TA EMPMAST 0 IS SUBJECT A
DBD0015 I DTE TA ADDR 0 IS SUBJECT B
DBD0019 I RELATIONSHIP ESTABLISHED UNDER PRIMARY NAMES
DBD1701 I DATA HAS BEEN ADDED TO RELATIONSHIP

```

Note: Compare with the example given for the same segment definition with interactive display forms in Figures 7  
Figure 5. Defining a Segment with Update Commands

**Exemple d'instructions du langage.**



## REMORA.

Système issu d'un projet universitaire animé par C. Rolland, REMORA vise à assister l'analyste dans la description de son problème dans une forme aussi indépendante que possible des moyens à mettre en oeuvre (= conception) et propose une solution partiellement automatisée, dirigée par un automate (pilote); jugeant irréaliste une approche complètement automatisée comme celle de ISDOS.

La grille d'analyse que nous avons retenue pour décrire un système de documentation est reprise comme telle par REMORA au titre de "ressources d'un processus d'aide à la conception - Réalisation d'un système d'informations" : des modèles, des langages, des méthodes et des outils.

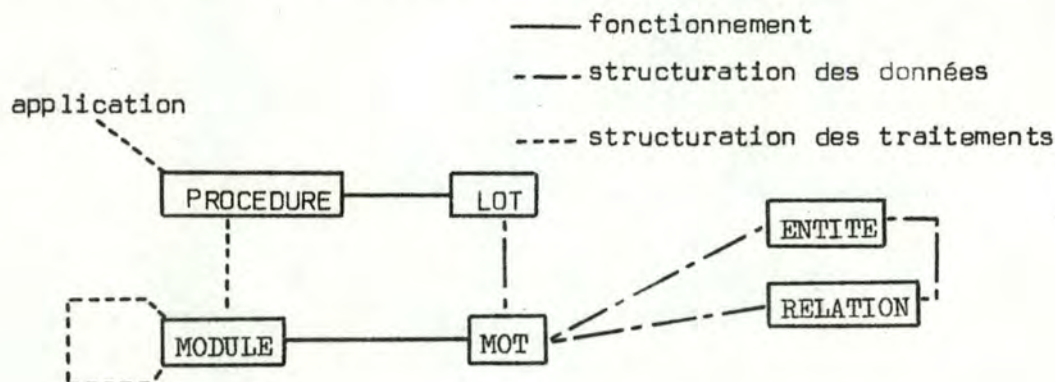
### METHODE

Contrairement à notre vision des choses, REMORA estime qu'il est indispensable de proposer une méthode - moyen de faire - qui guide l'utilisateur, assure l'ordonnancement et précise la nature des problèmes à résoudre. Ainsi préconise-t-il une analyse en 3 étapes :

- dans le cadre d'une procédure, description des messages qu'elle produit (les résultats ou besoins sont donc supposés connus)
- sélection des résultats, définition de leur structure et, par conséquent, de celle des traitements (Cfr langage)
- complément d'informations sur les "mots" utilisés par les traitements.

### MODELE

REMORA a pour objectif de réaliser le passage d'une procédure - conceptuelle - à un algorithme programmé - expression formalisée de la procédure, exécutable par ordinateur. Aussi à chacun des objets ou concepts du modèle conceptuel, auquel nous restreignons la présentation du système Remora, correspond leur équivalent au niveau de ce que nous avons convenu d'appeler la "Représentation". Nous ne présenterons pas ces "équivalents" et, en ce qui nous concerne, ce modèle s'architecture de la façon suivante :





Ces différents éléments sont notamment caractérisés par

- un identificateur qui, si l'on excepte les mots, vérifie la règle d'unicité
- une désignation ou un titre
- un résumé ou un commentaire.

#### ° Fonctionnement

Ce sont les concepts de PROCEDURE et de MODULE ainsi que les liens qui les unissent respectivement aux LOTS et aux MOTS qui doivent normalement permettre de décrire le fonctionnement du système.

La PROCEDURE est, en effet, la traduction en termes du modèle de l'évènement et des transformations qu'il engendre (sur les objets et leurs associations) indispensables au fonctionnement de l'entreprise.

tandis que

le MODULE traduit une définition ou une expression de ces transformations.

Ce qui finalement les distingue, ce sont les "liens d'utilisation" qui peuvent les unir soit à des LOTS s'il s'agit d'une PROCEDURE soit à des MOTS s'il s'agit d'un MODULE.

Et un LOT d'informations n'est, en fait, que la représentation d'un message décrivant les échanges entre le système d'informations et l'environnement qui l'alimente ou l'utilise, alors que le MOT, à ce niveau, ne doit être vu que comme une rubrique élémentaire ou une variable.

Il semble, toutefois, peu aisé de décrire le mode d'intervention d'une donnée (input, output,...) dans un traitement puisqu'

- au niveau des LOTS on ne peut guère que spécifier qu'il s'agit d'une structure d'édition ou non,
- et qu'au niveau des MOTS, si l'on peut signaler sa "nature" (input, output, intermédiaire), c'est vis-à-vis de l'APPLICATION et non du MODULE.

D'autre part, si l'on retrouve une volonté de tenir compte de l'aspect dynamique du fonctionnement et on en veut pour preuve la définition d'une procédure, on comprend mal pourquoi REMORA n'a pas élevé l'"EVENEMENT" (notion plus riche que celle, plus habituelle, de fréquence ou périodicité) au statut d'objet à part entière plutôt que de le considérer comme une caractéristique supplémentaire d'une PROCEDURE.

#### ° Structuration

Au niveau des traitements, l'APPLICATION est composée de familles de PROCEDURES appartenant à un même domaine d'activités de l'entreprise et traitant d'informations concernant une même population.

La PROCEDURE est à son tour constituée de plusieurs MODULES et les MODULES (maîtres) contiennent éventuellement d'autres MODULES (esclaves).



Mais ce qui fait l'originalité de ce modèle, c'est d'avoir récemment adapté celui-ci à partir des recherches en conception de bases de données pour offrir une plus grande souplesse de structuration des données.

L'ENTITE est la représentation d'un objet (clients, commandes...) du monde réel. L'OBJET est le constituant qu'on perçoit comme ayant une existence propre et dont la connaissance - nous dirons information - présente un intérêt qui justifie que le système d'information en donne une représentation.

tandis que

la RELATION représente une association d'objet qui permet d'envisager des situations complexes dans lesquelles chacun joue un rôle particulier.

Et c'est le MOT qui assure la jonction entre les 2 aspects de structuration (entités, relations) et du fonctionnement (lots et procédures, mots et modules) puisque le MOT est

- un élément d'un LOT (ou plusieurs) et à ce titre manipulé par un MODULE
- mais est aussi la représentation d'une propriété que l'on reconnaît à un objet ou une association.

#### ° Représentation

Ce modèle conceptuel dispose de certains concepts pour préparer la Représentation tels le format des MOTS, des renseignements de structuration d'un état imprimé,...

### LANGAGE

Trois grands types de descripteurs sous forme tabulaire - correspondant aux 3 étapes de l'analyse - rendent possible l'expression d'une solution à partir des concepts du modèle.

Des descripteurs de documents (LOTS) permettent de définir le contenu (MOTS) et la structure-physique- des messages produits par une PROCEDURE.

Mais le gros effort de Remora nous semble porter sur la définition d'un "langage de conception" destiné à décrire les traitements. Le raisonnement associé à ce descripteur des traitements est essentiellement déductif et basé sur une typologie des MOTS qui conduit l'analyste à définir un résultat à partir d'intermédiaires qui devront à leur tour être définis (arborescence). Ceci repose sur l'affirmation que la nature des traitements est étroitement liée à celle des résultats, classés en 3 catégories : conditionnels, inconditionnels ou liste de résultats auxquels correspondent les opérations conditionnelles, inconditionnelles et itératives, que le langage permet de prendre en considération.

Des descripteurs de MOTS sont également à la disposition de l'analyste pour compléter les renseignements que l'on en a.



## OUTIL

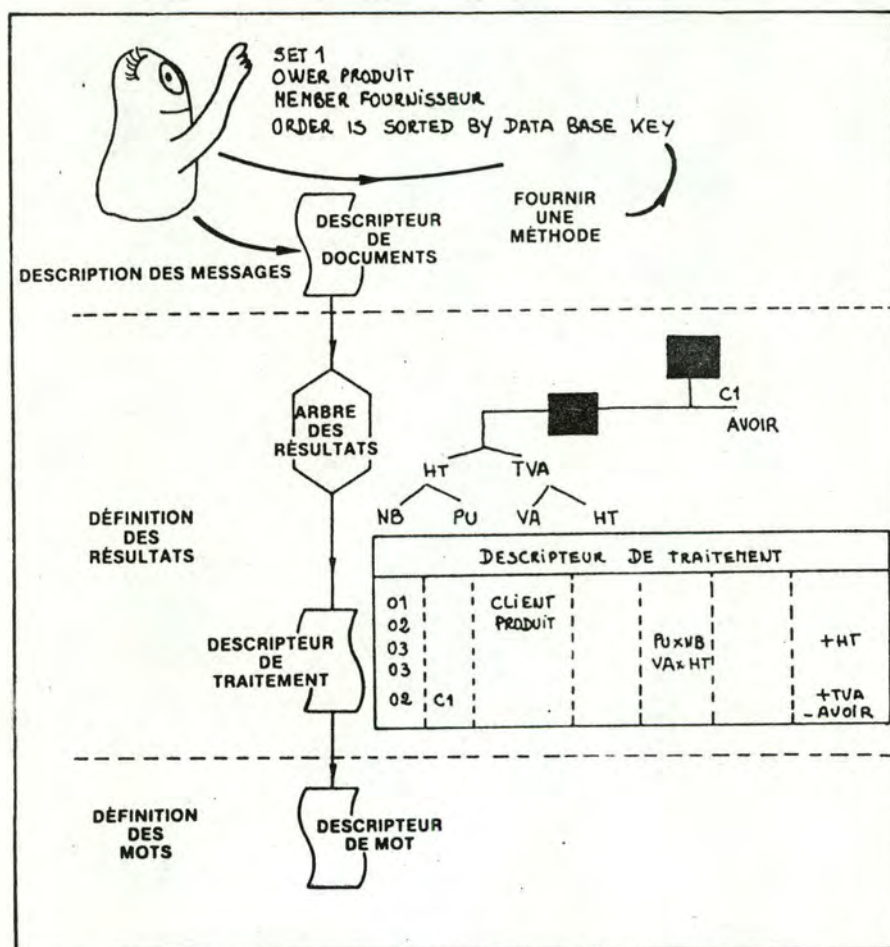
C'est à REMORA que nous avons emprunté, dans ses grandes lignes, la découpe d'un outil d'aide à la conception en fonctions de documentation de contrôle de cohérence et d'intégration et cependant nous avons très peu de renseignements sur les moyens mis en oeuvre pour les assurer, dans le cas de REMORA.

Tout au plus pouvons-nous dire que la fonction de documentation est assurée, dès lors qu'on sait que

- le stock documentaire, constitué du modèle conceptuel, est géré sous forme de base de données - Socrate
- le langage de requête de Socrate et un macro-langage permettent d'interroger ce stock
- des outils de dépouillement des descripteurs (compilateur) permettent de créer et mettre à jour ce stock.

Notons enfin que REMORA attache beaucoup d'importance, dans le cadre de la fonction d'intégration, à la détection des polysèmes et des synonymes de MOTS. (par le recours à des mots-clés, par exemple).

Ce qui singularise notamment REMORA par rapport aux autres systèmes, c'est d'avoir introduit les notions d'objet et d'association d'objets dans son modèle. Nous avons découvert ces notions par ailleurs et entre autres dans (Sundgren, 1975) et (Benci, Bodart, Bogaert, Cabanes, 1976) auxquels le lecteur se référera pour trouver la justification d'une telle structuration des informations.





## 2.6

**CASCADE.**Computer-aided system construction and documentation environment.

Ce projet (Université de Trondheim, Norvège) a pour objectif de

- développer une méthode de conception et de réalisation des systèmes d'informations
- créer les outils, en particulier pour assurer la fonction documentaire, pour faciliter le travail des concepteurs et autres analystes.

Toutefois, à l'heure actuelle, CASCADE limite son champ d'investigations à ce que nous avons convenu d'appeler la phase d'étude; d'autre part, le fondement théorique de ce projet, et notamment le modèle retenu, repose essentiellement sur les travaux de Langefors.

Enfin, dans le cadre du processus d'élaboration d'un système d'informations, CASCADE insiste tout particulièrement sur la collaboration indispensable des processeurs humains (analystes,...) et des processeurs automatisés (programmes pour mémoriser, maintenir à jour, présenter et analyser la documentation du système). La position adoptée est donc très proche de celle retenue par REMORA, si ce n'est que dans ce dernier projet le pilotage du processus était confié aux automates.

**METHODE**

Malgré notre volonté de ne pas prendre les méthodes en considération dans notre étude, il est indispensable de brièvement présenter la philosophie générale d'une telle méthode puisque certains concepts ou objets du modèle en sont directement issus.

A partir d'une "Structure de contrôle" supposée établie précédemment et de résultats ou besoins, par conséquent, connus et identifiés, le processus d'analyse va consister dans l'alternance

- d'analyse d'antériorité (PRECEDENCE) qui vise à détecter les relations d'antériorité entre les ensembles d'informations du système. Une telle relation est établie entre 2 ensembles d'informations (indépendamment de leur degré de décomposition) lorsque l'un est utilisé comme entrée par un processus de traitement qui produit l'autre.
- d'analyse de décomposition (COMPONENT) qui consiste à décomposer progressivement les ensembles d'informations en entités plus élémentaires et donc de dégager la structure de cet ensemble.

Et c'est seulement une fois en possession du graphe complet d'antériorité, tous les ensembles d'informations réduits à leur expression élémentaire, que l'on envisagera de

- décrire les processus de traitement correspondant aux relations d'antériorité
- et d'éventuellement réaliser certaines synthèses en regroupant certains processus ou certains ensembles d'informations (élémentaires).



## MODELE

Ce modèle, l'un des plus riches que nous ayons abordés, a pour principale caractéristique de clairement dissocier - au niveau des concepts - le fonctionnement de la structuration du système, tout au moins en ce qui concerne les données. ("information set" et "concept").

### ° Fonctionnement

Dans CASCADE, le fonctionnement d'un système d'informations est décrit au moyen

- des 3 types d'objets suivants :

PROCESS (ou SUBSYSTEM) ou processus traduisant une relation d'antériorité

INFORMATION SET ou ensemble d'informations, de taille quelconque et

SIGNAL ou événement déclencheur qui permet de prendre l'aspect dynamique du fonctionnement en considération.

- et des types de relations pouvant exister entre ces objets :

PRECEDENCE : lie un INFORMATION SET à un autre que le "precede", c.à.d. à partir duquel il est produit.

SUCCEDECE : est la relation inverse

INPUT : relie un INFORMATION SET à un PROCESS qui l'utilise comme entrée

OUTPUT est la relation inverse et lie un PROCESS à un INFORMATION SET produit par ce PROCESS

ENTRY unit un SIGNAL à un PROCESS, activé par la survenance de ce SIGNAL

EXIT est la relation inverse et relie un PROCESS à un SIGNAL, armée par une activation (mise en oeuvre) du PROCESS.

### ° Structuration

Au niveau des traitements, la décomposition hiérarchique de ceux-ci se fait apparemment sur base d'une codification de leur identifiant (SHORT NAME).

La structuration des données, quant à elle, est assurée par les notions :

- d'ELEMENTARY-CONCEPT : ou type d'informations à propos d'une propriété d'un type d'objet à un moment donné. Une propriété doit être vue comme une caractéristique propre ou une relation avec d'autres objets.
  - de PRO-CONCEPT, identifie en cours d'analyse la décomposition, correspond à un conglomérat d'E-CONCEPT ou d'autres PRO-CONCEPTS progressivement identifiés et décomposés.
  - de CONSOLIDATED-CONCEPT : ou ensemble reconstruit d'E-CONCEPT, avec plus d'une propriété à propos d'un seul (type d') objet,
- reliés entre elles par les relations de COMPONENT et d'ASSEMBLY (inverse).



Ce qui fait l'originalité de ce modèle, outre cette différence entre PRO-CONCEPT et C-CONCEPT suivant qu'on est en phase d' "EXPANSION" ou de "CONTRACTION" (synthèse), c'est la jonction entre le fonctionnement et la structuration, assurée par la relation DECLARATION reliant un INFORMATION-SET à un CONCEPT.

Ceci repose sur la constatation que plusieurs INFORMATION SET peuvent être décrits par le même ensemble de composants : constatation qui justifie cette séparation entre la description de l'INFORMATION SET et l'INFORMATION SET lui-même, dont le degré de détail de la description est reflété par les 3 différents TYPES de CONCEPTS. Cette solution semble plus riche que celle retenue par Minos qui résolvait le même problème par une codification, au niveau des STRUCTURES LOGIQUES, de l'identifiant (RACINE + niveau d'ELABORATION).

#### ° Représentation

Ce modèle permet de préparer le travail de REPRESENTATION des E-CONCEPTS ou C-CONCEPTS en donnant la possibilité d'identifier les TERMS associés à un CONCEPT :

- ceux qui identifient l'objet : IDENTIFIER TERM (et TRANSACTION NUMBER)
- ceux qui désignent la (E-) ou les (C-) propriétés : NAME TERM, VALUE-TERM
- et ceux qui localisent celle(s)-ci dans le temps : TIME TERM.

Un élément dans ce modèle est donc repéré et caractérisé par

- un SHORT NAME ou identifiant primaire
- un 'LONG NAME' correspondant à une désignation complète
- un TYPE indiquant son appartenance à une classe (PROCESS, SIGNAL, INFORMATION SET, CONCEPT, TERM, auquel il faut ajouter ANALYST, c.à.d. responsable d'une partie du système en construction) et à une sous-classe en ce qui concerne les CONCEPTS (E-, C-, PRO-) ou les TERMS.
- un ensemble de RELATIONS
- et les objets touchés par ces RELATIONS.

## LANGAGE

A chacune des caractéristiques précédentes correspond un "opérateur"

- SHORTNAME
- LONGNAME
- SUBSYSTEM, INFOSET, CONCEPT, etc
- PRECEDENT, INPUT, DECLARATION,...

A partir de là, 4 commandes suffisent pour créer (NEW), modifier (CHANGE), supprimer (DELETE) ou obtenir (GET) une liste de valeurs (opérandes) d'un opérateur dans le stock documentaire. Voici quelques exemples de commandes :

#### Typical DOC-commands

<u>Command</u>	<u>Remark</u>
GET P(A):	Retrieves precedents of A
NEW I (SUB1): C, D, E	{ Definition of SUB1's input/ output
NEW O (SUB1): A, B	
NEW D (A): 'PROD-PLAN'	A is declared to have the (global) concept 'PROD-PLAN' (long name)



D'autre part, une particularité de ce langage est d'autoriser des expressions compactes et denses en renseignements sous forme soit de macro

une macro, reprenant plusieurs commandes, peut être définie (DO MACRO identificateur ), avec éventuellement des paramètres formels, actualisés lors d'une exécution (DO identificateur ) de la macro.

soit d'instruction composée  
du genre de :

NEW SUBSYS. INPUT/INFOSET (SYS1): INFO1, INFO2

3 new identifiers will be stored in database: SYS1 of type SUBSYS, INFO1 and INFO2 of type INFOSET. The relation: INPUT of SYS1 is INFO1 and INFO2, will be stored.

dont le format est le suivant :

<composed order> :: = <elementary command><operatorlist1>  
[ /<operatorlist2> ] (<operandlist1>):  
[ <operandlist2> ]

et qui signifie :

<elementary command> : defines the kind of operation on the database  
= NEW (data to be included in database)  
= GET ( data from database)  
= DELETE ( data in database)  
= CHANGE ( data in database)

<operatorlist1> : contains operators to be applied on each element  
in <operandlist1>  
= <type value> e.g. INFOCONCEPT  
= <rel. value> e.g. PRECEDENT  
= <rel. operator> e.g. NAME, RELATION

<operatorlist2> : contains operators to be applied on each element  
in <operandlist2> or on identifier names  
produced by <operatorlist1> and <operandlist1>  
= <operatorlist1>, but without <rel. value>

<operandlist1> = list of identifier names and <simple get order>

<operandlist2> = list of identifier names and <simple get order>,  
or one <value of type>

<simple get order> :: = <relation value> (<id. list>)

## OUTIL

- ° Fonction de documentation, - CASCADE reconnaît explicitement qu'un système de documentation doit au minimum
- posséder une représentation du modèle sous forme d'une Base de données
- avoir un langage de communication et les outils de traduction afférents
- disposer de "facilités de présentation du système documenté".



Concrètement il s'agit d'une série de rapports ou de graphes :

- ° documentation de tous les objets (ou éléments) du stock
- ° tables des INFORMATION SETS et des CONCEPTS correspondants
- ° structure hiérarchique des CONCEPTS (documentation de l'analyse de décomposition)
- ° structure d'antériorité (précédence) des INFORMATION SETS
- ° une série de 6 diagrammes possibles de fonctionnement.

° Fonction d'intégration. - Le système veillera à faire respecter la règle suivante lors de toute mise à jour : les identifiants ou SHORT NAME sont uniques excepté pour les INFORMATION SETS qui sont qualifiés par le SYSTEM ou PROCESS avec lequel ils sont en relation et au sein duquel cet identifiant est unique.

° Fonction de contrôle de cohérence. - Nous ignorons si CASCADE dispose de pareils outils.

De CASCADE, nous retiendrons essentiellement

la richesse de son modèle qui distingue bien le fonctionnement,  
la structuration et la représentation  
la souplesse de son langage.



**CADIS.**

Computer-aided design of information system.

Fort proche de CASCADE, ce projet (Institute of Technology, Stockholm) propose un modèle et une méthode d'analyse sinon identiques du moins fort semblables à ceux présentés dans le projet CASCADE.

Mais ce qui différencie CADIS de CASCADE et de tous les autres projets, c'est assurément la souplesse de son outil de documentation, non orienté vers une philosophie particulière ou un modèle déterminé d'un système d'informations.

Aux yeux des concepteurs du projet CADIS, un outil de documentation doit être capable

- d'être utilisé pour décrire un système d'informations lors des différentes phases de son élaboration et à différents niveaux de détail
- de s'adapter aux évolutions des idées en matière de systèmes d'informations
- de prendre en considération les particularités et éventuellement le modèle propre à une organisation
- d'accepter plusieurs formes d'entrées et de proposer différentes techniques de présentation de la documentation
- d'exercer certains contrôles pour garantir la cohérence du système en construction.

A partir de ces spécifications - sévères, CADIS propose, en fait, un software de gestion de bases de données-relationnelles - et notamment

- son langage de description de données, pour décrire le modèle adopté que l'on suppose établi par ailleurs
- son langage de manipulation de données, pour mettre à jour et interroger la base de données.

Si l'on se rappelle que CADIS est proche de CASCADE par son modèle et sa méthode, et qu'il s'en distingue par son outil, on comprendra aisément que nous n'avons pas suivi notre grille d'analyse habituelle mais que nous avons privilégié un tant soit peu la présentation de l'outil.

**OUTIL**

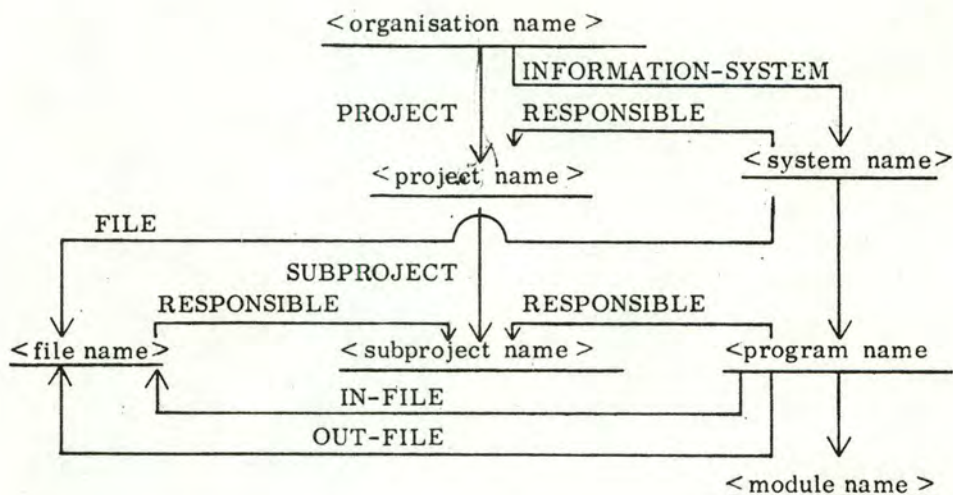
Dans CADIS, la base de données est construite à partir de triplets relationnels du genre (X, Y, Z) où "Y relie Z par la relation binaire z", comme par exemple

(INPUT, SYS 1, BCDE) : BCDE est un infoset input du système SYS 1

(DECLAR, BCDE, DBON) : DBON est un concept déclarant un infoset BCDE



Voici un exemple d'une partie d'un modèle et de ses triplets relationnels



On peut toutefois se demander si le fait de mettre sur le même pied des attributs (LONG NAME, system name, longname) et des relations réelles d'objets (INPUT, system name, infoSetName) n'est pas une présentation trop artificielle pour l'utilisateur ou l'analyste.

Langage de manipulation. Tous les "faits" à propos du système en construction sont donc communiqués à CADIS et manipulés par lui sous forme de "Triplet relationnel". Une entrée normale de CADIS est donc constituée

- d'une commande (MAKE, CHANGE, DELETE et GET notamment)
- d'un triplet.

Sauf en interrogation, les 3 membres du triplet doivent être spécifiés; par contre, en interrogation, les membres sur lesquels portent la "recherche" seront évidemment manquants et transmis par le système CADIS.

```

1. *OK.NEXT
  ?/MAKE
  ?PROJECT
  ?ELECTRO-CADIS LTD
  ?MIS-720B
  *NEW IDENTIFIERS
  *PROJECT
  *ELECTRO-CADIS LTD
  *MIS-720B
  
```

```

2. *OK.NEXT
  ?/GET
  ?PROGRAM
  ?INVENTORY-CONTROL
  ?:
  
```

Et le système (ordinateur) répond par exemple:

```

*PROGRAM OF INVENTORY-CONTROL IS EXP-ORDER
*INVOICING
*ORDER-SORTING
  
```



Enfin 2 particularités du langage offrent aux utilisateurs une puissance accrue d'interrogation :

- une commande (PREC) permet d'obtenir du système des structures entières en précisant dans une seule commande le chemin à suivre dans la base (pour obtenir par exemple la hierarchie pro-concept, e-concept, term d'un élément).
- un générateur de rapports restitue les informations souhaitées, présentées dans un tableau dont on a dû au préalable - dans la commande - spécifier le format ou l'image.

#### Exemple du générateur de tableaux

OUTPUT FORMAT

?PROGRAM,MODULE,IN-TERM,OUT-TERM

MODULE-LENGTH

?18,13,12,12

THE TRIPLE

?MODULE

?:

?:

Et l'écran affiche par exemple:

PROGRAM	MODULE	IN-TERM	OUT-TERM
EXP-ORDER	SALES-CALC	INV-QTY ORD-QTY REORD-POINT	INV-QTY EXP-QTY
INVOICING	?	?	?
ORDER-SORTING	?	?	?

Pour terminer signalons que CADIS envisage de prendre en considération la fonction-contrôle de cohérence, mais sous forme de contraintes d'intégrités définies sur les triplets de la base.

Sa souplesse fait de CADIS un bon outil de mise au point de modèles des systèmes d'information - à l'université par exemple ; car n'est-ce pas un instrument trop riche pour une entreprise qui s'est donné un modèle déterminé ? Sans y répondre, on peut se poser la question, surtout si ce software n'est employé que pour gérer la documentation.



## 2.8

**ISDOS.**

Information System design and optimisation system.

ISDOS est un projet ambitieux, dirigé par Teichroew (Michigan), qui s'est donné pour objectif d'automatiser le processus d'élaboration (conception et réalisation) d'un système d'informations.

La première étape du projet - et celle qui nous intéresse dans le présent travail - fut de développer un système automatisé de documentation et d'analyse des "exigences en informations" pour assister le "concepteur" dans les phases de conception (logique) du système.

**METHODE**

Comme la plupart des autres systèmes, l'approche d'ISDOS pour déterminer les "exigences" est d'identifier les OUTPUT (de les supposer établis) et puis de les décrire avec suffisamment de détails pour que les INPUTS, les PROCESSES et autres données mémorisées nécessaires puissent être identifiés, et à leur tour détaillés progressivement.

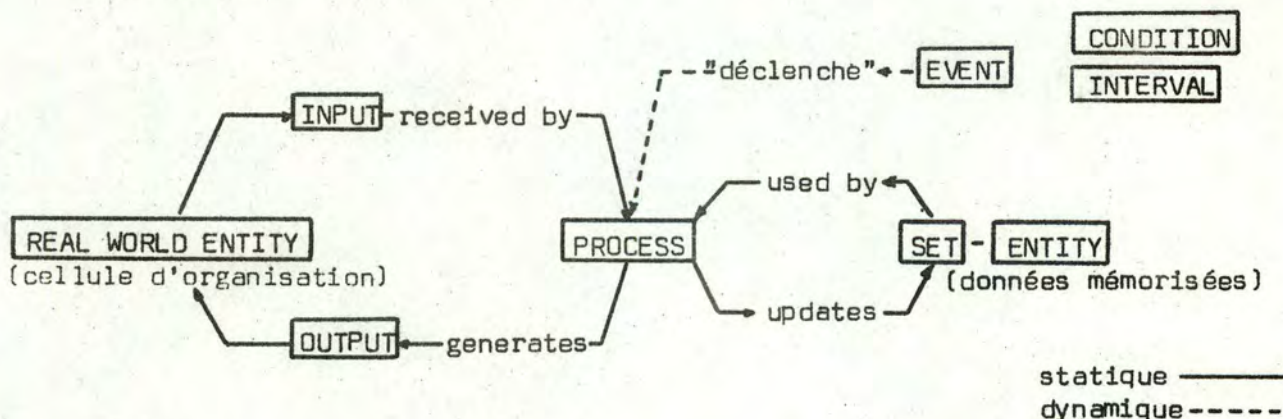
**MODELE**

ISDOS propose, bien sûr, un modèle assez complet prenant en considération tous les aspects - à un niveau conceptuel - de la description d'un système d'information :

- le fonctionnement - statique et dynamique - du système
- la structuration des traitements et des données
- les quelques renseignements indispensables à la représentation
  - ultérieure - du système à l'étude.

## ° Fonctionnement

A ce niveau, le modèle permet de décrire les éléments du schéma suivant :





### Aspect statique :

Par INPUT/OUTPUT, il faut comprendre les documents ou tout autre support d'information, saisie ou produits par le système d'informations.

Un PROCESS correspond à une manipulation de données selon une PROCEDURE qui regroupe les différentes formules, conditions et décisions nécessaires à l'obtention de certaines données.

Les notions de SET et ENTITY sont introduites pour tenir compte des données mémorisées - Base de données ou fichiers - par le système entre l'exécution de traitements.

Une REAL WORLD ENTITY est - vraisemblablement - une cellule d'organisation qui génère les INPUTS et utilise les OUTPUTS générés par le système d'informations.

### Aspect dynamique :

Cet aspect du fonctionnement est principalement pris en charge par le concept d'EVENT.

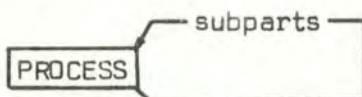
Un EVENT est causé par le début ou la fin d'un PROCESS, ou en toute généralité quand une CONDITION devient vraie ou fausse. Cet EVENT déclenche alors d'autres PROCESSES ou d'autres EVENTS. L'élément INTERVAL permet de décrire le comportement périodique de certains EVENTS.

Des indications de volume et de fréquence d'INPUT/OUTPUT complètent éventuellement la description dynamique du système d'information.

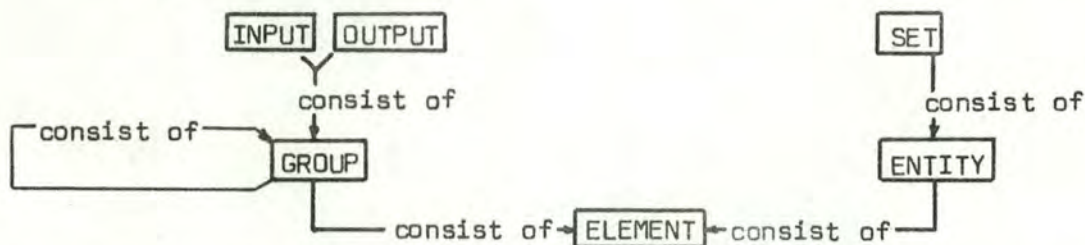
### ° Structuration

A ce niveau, le modèle autorise

- une décomposition -hiérarchique- des traitements ; la définition d'un PROCESS se fera par pas successifs jusqu'au niveau de détail souhaité (sous-processus).



- Une décomposition des données associées aux INPUTS/OUTPUTS aux données mémorisées
- Selon le schéma suivant :



Un ELEMENT est "la plus petite unité de donnée qui ait un nom unique et une valeur.

tandis qu'un GROUP est une combinaison d'autres GROUPS ou d'ELEMENTS pour tenir compte des différentes structures rencontrées.

Un SET doit correspondre à la notion de fichier

tandis qu'une ENTITY doit s'apparenter à celle de Record ou d'enregistrement.

(Signalons l'existence du concept de RELATION destiné à "caractériser les relations entre GROUP et ELEMENT").



## ° Représentation

La définition d'un ELEMENT sera complétée par quelques renseignements spécifiques aux nécessités de la représentation et notamment par  
un FORMAT  
un ensemble de VALUES possibles  
et éventuellement par des SYNONYMS

## LANGAGE

### ° Objectif :

Le PSL (Problem Statement Language) a pour mission de permettre à l'utilisateur, éventuellement assisté de l'informaticien, de définir précisément son problème à partir des concepts du modèle. Ce langage se veut donc

- le plus naturel possible
- autorisant une approche structurée (TOP DOWN) de la définition du problème
- susceptible d'accroître les capacités de l'analyse lorsqu'il détermine ses exigences en information.
- permettant un contrôle de cohérence du problème tel qu'il est défini.

De plus, il doit être à ce point formalisé qu'il puisse être "analysable" par programme (et pas seulement "lisible" en vue d'une simple mémorisation).

Concrètement, comment ces spécifications sont-elles prises en considération ?

### ° Solution retenue :

L'unité de base du PSL est la SECTION, correspondant à <sup>d'un des</sup> la description ELEMENTS principaux du modèle, au sein de laquelle on peut non seulement préciser les caractéristiques propres de cet ELEMENT mais aussi définir les relations avec les autres ELEMENTS (SECTIONS) du système.

Ces SECTIONS sont décrites au moyen de phrases le plus proche possible de l'anglais courant et l'ordre d'agencement des SECTIONS et des phrases au sein des SECTIONS est assez libre, offrant une réelle souplesse à l'utilisateur.

## OUTIL

Le PSA (Problem Statement Analyzer) est un outil (software) très complet dans l'état actuel de l'art et qui assure les fonctions

- de documentation et d'intégration
  - en acceptant et analysant les phrases du PSL
  - en maintenant à jour une Base de données des éléments définis en PSL
  - en restituant tous les renseignements demandés par un utilisateur
- de contrôle de cohérence
  - en analysant les relations ayant trait
    - ° à la structuration des données
    - ° au fonctionnement statique et dynamique du système
  - et en produisant les rapports de ces analyses.







d'analyse dynamique a pour objectif de vérifier si les données nécessaires à la génération des outputs seront disponibles au Moment voulu.

PSA comprend également d'autres modules, entre autres de préparation des phases suivantes de réalisation (volumes,...) que nous n'avons pas repris dans le cadre de ce trop bref exposé.

```

PROCESS SALES_ORDER_PROCESSING;
  PROCEDURE:
    IF INVENTORY RECEIPT TRANSACTION THEN INVENTORY RECEIPT TRANS_PROC
    ELSE IF CUSTOMER ORDER TRANSACTION THEN CUSTOMER_ORDER_PROC
    ELSE IF END OF FILE TRANSACTION THEN END_OF_FILE_PROC
    ELSE ILLEGAL TRANSACTION CODE;
  USES SALES_ORDER_PROC HEADERS;
  RECEIVES SALES_ORDER_PROC INPUT;
  SUBPARTS ARE INVENTORY_RECEIPT_TRANS_PROC, CUSTOMER_ORDER_PROC, END_OF_FILE_PROC,
  ILLEGAL_TRANSACTION_CODE;

PROCESS INVENTORY_RECEIPT_TRANS_PROC:
  USES NUMBER_RECEIVED TO UPDATE NUMBER_ON_HAND;
  UPDATES NUMBER_OR_INVENTORY_RECEIPT_TRANS;
  USES NUMBER_RECEIVED TO UPDATE TOTAL_NUMBER_RECEIVED;
  GENERATES INVENTORY_RECEIPT_TRANS_OUTPUT;
  PROCEDURE:
    ADD NUMBER_RECEIVED TO NUMBER_ON_HAND
    ADD 1 TO NUMBER_OF_INVENTORY_RECEIPT_TRANS
    ADD NUMBER_RECEIVED TO TOTAL_NUMBER_RECEIVED;

INPUT SALES_ORDER_PROC INPUT;
  CONSISTS OF TRANSACTION_CODE, TRANS_PARM;
  RECEIVED BY SALES_ORDER_PROCESSING;
  ELEMENT TRANSACTION_CODE;
  VALUE 0 THRU 9;
  ELEMENT TRANS_PARM;
  SYNONYMS ARE NUMBER_ORDERED, NUMBER_RECEIVED;
  VALUE 0 THRU 9999;

ENTITY SALES_ORDER_PROC SUBHEADERS;
  CONSISTS OF HEADER_NOH, RECEIVED_HEADER, SHIPPED_HEADER, CHECK_NOH_HEADER,
  CHECK_ORDER_HEADER;
SET SALES_ORDER_PROC HEADERS;
  CONSISTS OF SALES_ORDER_HEADER, SALES_ORDER_PROC_SUBHEADERS;
ENTITY SALES_ORDER_HEADER;
  CONSISTS OF SECTION_1_HEADER, SECTION_2_HEADER;

```

Quelques exemples d'instructions en P.S.L.



## 2.9 PROLONGEMENTS POSSIBLES...

### 29.1 ...VERS UNE AUTOMATISATION...

A partir des exigences en informations des utilisateurs, certains systèmes se sont fixés comme objectif d'automatiser le processus d'élaboration du système et de directement fournir les programmes et organisation de Bases de données ou de fichier, réduisant les interventions humaines au minimum.

Le système PAC de la CGI a cet objectif et le réalise partiellement, mais l'homme continue d'intervenir lors des phases de choix des moyens et de construction physique. Nous avons préféré présenter dans ses toutes grandes lignes l'un des systèmes plus ambitieux de ce point de vue : Il s'agit d'ISDOS dont nous avons déjà expliqué le PSLangage et le PSanalyser.

A partir de la définition complète et cohérente du problème, fournie par PSL et PSA, ISDOS se propose d'automatiser - au maximum - les phases

- de choix des moyens
- d'architecture physique
- et de construction - programmation
  - organisation physique détaillée des données

en faisant appel à l'homme pour des compléments d'information (hardware disponible par ex.), mais non pour lui demander de résoudre des problèmes.

Un premier module (SODA) va <sup>(°)</sup>

- générer des alternatives de configurations-hardware (CPU, taille mémoire) et d'architectures physiques du système (structures de programmes, structures de données...)
- choisir la meilleure configuration possible et optimiser - ou en tous cas, évaluer les performances - les architectures pour en sélectionner une.
- produire les résultats suivants :
  - . liste des ressources (moyens) nécessaires
  - . spécifications des programmes à générer
  - . spécifications des structures de fichiers et des supports physiques nécessaires
  - . une proposition de planning (scheduling des tâches) d'exécution des programmes pour rencontrer les exigences des utilisateurs (essentiellement à partir des "événements-déclencheurs").

Ensuite, un autre module (DATA REORGANISER), à partir des spécifications de SODA (supports, organisation souhaitée) et de la définition des données de PSL (contrôlées par PSA), va réellement mémoriser ces données sur des supports physiques suivant les modalités souhaitées.

D'autre part, un dernier module (CODE GENERATOR) va réellement générer les programmes se basant sur les spécifications de SODA et les PROCEDURE du PSL et en respectant les interfaces spécifiées par le DATA-REORGANISER.

Enfin, un module directeur (SYSTEM DIRECTOR) accepte le code généré, les spécifications de la planification et celles du DATA REORGANISER produit le système-cible complètement construit. Ce système est prêt à recevoir les INPUTS de l'environnement et produire les OUTPUTS requis par l'utilisateur et suivant ses exigences notamment de timing.

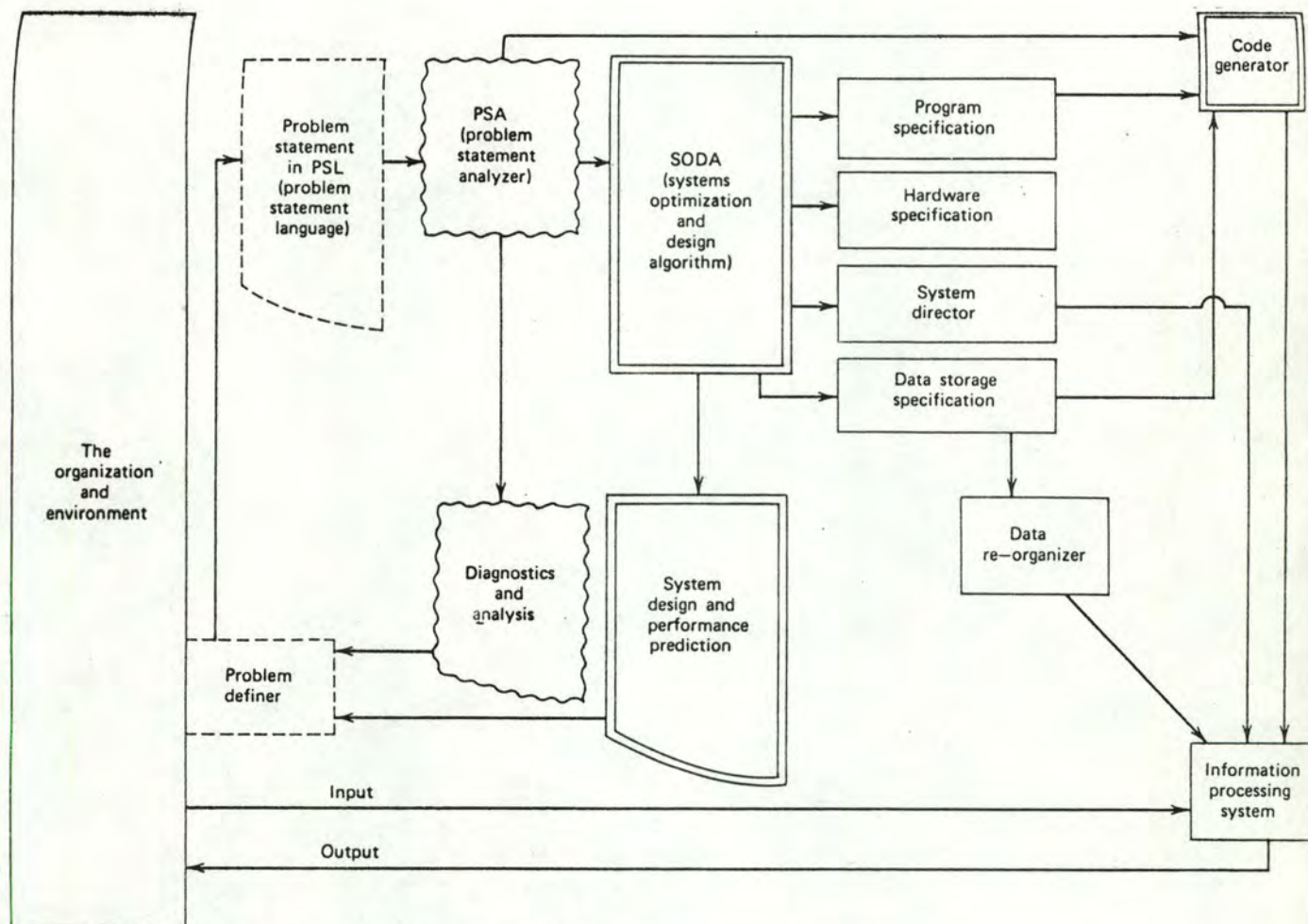
---

(°) voir page suivante



Il est certain que, plus que pour tout autre système, cette automatisation n'est faisable que si l'on distingue très précisément les phases de CONCEPTION et de REALISATION, sans faire la moindre hypothèse d'organisation physique dans la première phase.

D'autre part, insistons avec Teichroew sur la nécessité qu'il y a de maintenir la même souplesse d'utilisation en cas de modifications ou changements des exigences de l'utilisateur que lors de la création initiale du système. Ce n'est qu'à cette condition qu'un tel système sera pleinement justifié, rentable et susceptible de rendre de grands services aux entreprises.



Information System Design and Optimization System (ISDOS).

(°) Ce module - qui repose sur les travaux de Langefors - a été conçu à une époque où certains concepts actuels (bases de données...) n'avaient pas encore vu le jour. Nous pensons cependant que cela n'enlève rien à la philosophie générale d'une telle démarche et qu'à ce titre celle-ci méritait d'être reprise comme un prolongement possible d'un système de documentation.

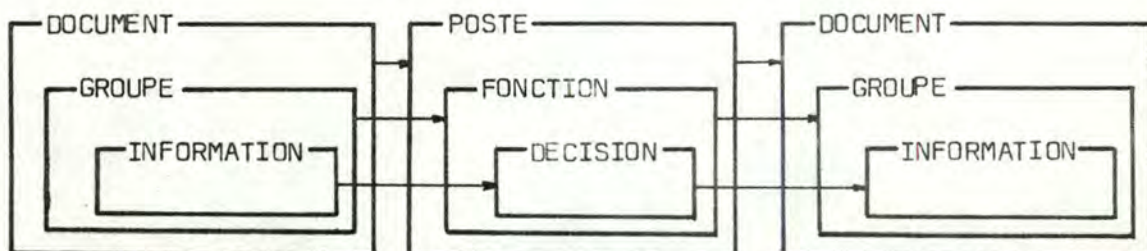


## ...VERS UNE AIDE ACCRUE AU GESTIONNAIRE

Un autre type de prolongemens possibles est mis en évidence par (Bourgeois, 1977) qui, à partir d'un système documentaire assez rudimentaire, a pour objectif de dégager la "STRUCTURE DE CONTROLE" de l'organisation, à usage donc du gestionnaire et non plus de l'informaticien.

Le modèle du système documentaire permet une description du fonctionnement du système d'informations et une structurations des éléments par

- l'inventaire des postes - cellules homogènes de traitement de l'information
- des documents émis et reçus par ces postes
- la décomposition de ces 2 éléments de la façon suivante :



Le langage est constitué de 2 types de descripteurs sous forme tabulaire : l'un pour décrire les postes, l'autre pour les documents, tandis que la fonction documentaire de l'outil est assurée par la saisie-compilation - de ces descripteurs, la tenue de 2 fichiers correspondant aux 2 types de descripteurs et par l'émission d'un certain nombre de rapports.

L'originalité vient de ce que le système dispose d'un outil supplémentaire - simple - qui aide l'analyste à détecter les "BOUCLES" constituées par des successions alternées de DECISIONS et d'INFORMATIONS, à examiner les durées - ou DELAIS - de ces boucles (à partir de la périodicité des traitements et de la durée des transmissions).

Ceci permet donc de connaître les INFORMATIONS (DOCUMENTS) et les DECISIONS (POSTES) qui contribuent effectivement au contrôle et de détecter les circuits morts d'informations, les circuits interrompus, les phénomènes de ping-pong, les circuits multiples, les délais excessifs et les documents inutiles.

D'autres automatés ont pour mission d'assister l'analyste - de gestion - dans son travail de transformation de la "Structure de boucles" précédente en une "structure de contrôle efficace et légitime" sous forme d'un modèle dont les éléments soient des variables mesurables (de mouvement ou d'état) et les relations, des expressions mathématiques contenant ces variables.

La présentation - trop - brève de ce système avait pour seul objectif de montrer un type d'apport possible d'un outil de documentation à la gestion - au sensle plus noble du terme - d'une organisation.



3.

### **PROPOSITION d'un SYSTEME de DOCUMENTATION.**

La première partie de ce travail a - notamment - présenté les grandes spécifications d'un système de documentation tandis que, dans la seconde partie, nous avons examiné comment différents systèmes existants prennent en charge ces spécifications.

Nous allons tenter, dans cette dernière partie, de proposer la terminologie ou le modèle d'un système de documentation ainsi que le langage qui permettent de manipuler les concepts de cette terminologie convenue, c'est-à-dire d'exprimer les exigences de l'utilisateur du futur système d'informations, au moyen de ce formalisme.

Car il n'est, peut-être, pas superflu de rappeler une dernière fois que nous ne prendrons en considération que des concepts manipulés par les activités de CONCEPTION auxquelles notre système est dédié.

Il resterait donc à construire les outils de documentation et de contrôle pour que ce système soit digne de ce nom et puisse réellement fonctionner et rendre des services.<sup>(°)</sup> On comprendra que le temps nous ait manqué dans ce travail pour envisager de présenter un outil opérationnel.

Il faut cependant noter que la façon dont nous avons conçu notre modèle (des objets, des attributs d'objets, des relations entre objets) permettrait assez facilement de déboucher sur une base de données physique, construite au moyen du SGBD choisi, tandis que, momentanément, on pourrait prendre le langage de requêtes ou de manipulation de ce SGBD pour profiter des possibilités de la terminologie adoptée en attendant d'introduire le langage, plus adapté à l'analyse, que nous proposons.

---

(°) puisque c'est délibérément que nous refusons de prendre en considération une quelconque méthode au niveau d'un système de documentation.



### 3.1 MODELE CONVENU.

Ce chapitre est essentiellement consacré à la définition des concepts retenus, aux conventions dont ils ont fait l'objet, et à l'examen des relations qui les unissent, du triple point de vue du "fonctionnement," de la "structuration" et de la "représentation".

Fidèle à ce que nous prônons par ailleurs et à ce que proposent les concepteurs de bases de données, nous n'avons tenu compte en aucune manière de considérations "d'implémentation" en élaborant cette "structure conceptuelle" dans laquelle l'on ne retrouvera uniquement des objets (SYSTEME, RESSOURCE, EVENEMENT, ENTITE, INFORMATION et DONNEE)

des relations entre ces objets (de fonctionnement, de structuration et de représentation)  
des attributs d'objet (RACINE, DESCRIPTION, TYPE...).

Il est important de souligner que l'intérêt porté à ces concepts n'est pas seulement didactique, mais que toute utilisation d'un système de documentation par les analystes ou les utilisateurs est conditionnée par une claire compréhension des concepts qu'ils seront amenés à manipuler et donc à assimiler sans équivoque, ni ambiguïté. Cette assimilation constitue probablement l'effort principal qui leur sera demandé pour profiter des facilités de l'assistance subséquente du système, véritable outil mis à leur disposition.

Les définitions retenues et leur coordination en vue de la constitution de cette "terminologie" sont à l'évidence issues de la compilation des auteurs consultés mais ont parfois subi une mise en forme et certaines conventions, impliquées par notre vision personnelle. Et cependant, ce modèle est loin d'être complet et exhaustif et certains objets, certaines relations ou certains attributs gagneraient assurément à être plus profondément analysés.



## 31.1

## FONCTIONNEMENT et STRUCTURATION des TRAITEMENTS

Pour dégager le premier concept, nous allons exploiter une remarque, peut-être passée inaperçue, faite dans la première partie et que nous rappelons rapidement :

"A des niveaux d'observation différents, ORGANISATION, UNITES-COMPOSANTES et ACTIVITES ou PROCESSUS de TRAITEMENT intègrent tous un certain nombre de ressources pour réaliser un objectif". Aussi leur avons-nous préféré, dans un premier temps, la dénomination unique de SYSTEME.

L'emploi d'un terme unique, sans qu'il faille tenir compte de la dimension et de la nature des entités ainsi regroupées, simplifie définitions et applications ultérieures, élimine les questions fastidieuses et souvent vaines de classification et renforce ainsi la souplesse d'utilisation du "système de documentation".

## 311.1

## SYSTEME

Un SYSTEME peut se définir comme la combinaison opérationnelle de ressources (humaines, physiques, financières et informationnelles) qui transforme les ressources-inputs pour réaliser l'objectif auquel les outputs répondent.

## systeme et hiérarchie

La réflexion qui va suivre aurait dû, en bonne logique, trouver sa place dans le paragraphe suivant puisqu'elle a trait à la STRUCTURATION des TRAITEMENTS. Nous avons toutefois estimé préférable de regrouper toutes les caractéristiques propres aux "SYSTEMES" en un seul endroit.

L'aspect général de la définition, reprise ci-dessus, en ce qu'elle s'applique à toutes les situations quelles que soient la taille du SYSTEME étudié et la structure de l'organisation, pré-suppose les notions de niveaux, de composition ou de hiérarchie de systèmes, qui font l'objet d'une abondante littérature.

Ainsi, par exemple, Lemoigne paraphrasant Simon :

L'effet arborescent est le processus par lequel nous parvenons à maîtriser la complexité (...) parce qu'il permet l'identification progressive de niveaux de quasi-indépendance de plus en plus simples au fur et à mesure que se poursuit le processus de décomposabilité.

(Lemoigne, 1974)

Mais si nous prenons en considération cette décomposition hiérarchique - qui n'implique pas une structure de subordination, nous nous refusons cependant de transposer ces différents niveaux dans une nomenclature.

D'autre part, quelle que soit sa grandeur, tout système est constitué d'éléments caractéristiques dont l'importance et la complexité croissent, bien sûr, avec la dimension du système mais qui sont fondamentalement identiques. Quels sont-ils ?

ture et une de ses rubriques élémentaires sont, dans notre esprit, 2 exemples de ressources informationnelles. D'autre part, le terme de ressource-PROPRE recouvrira souvent celui, moins déroutant, de "donnée mémorisée" (ou history).



## six éléments caractéristiques d'un système

Le premier de ces 6 éléments est le fait d'avoir une

### 1 - FONCTION

C'est l'objectif (Nadler, 1972) ou l'objet (Martzlozz) du système concerné, au sens d'un résultat à rechercher ou à produire.

Exemples : contrôler les comptes-clients pour un système "gestion du crédit-clientèle" ou éditer les factures pour un système "édition"facturation"...

Pour réaliser cette FONCTION, le SYSTEME doit intégrer des RESSOURCES de plusieurs TYPES : humain, physique, financier ou informationnel. Les 3 façons différentes dont une RESSOURCE peut intervenir dans un SYSTEME particulier sont à l'origine des 3 éléments caractéristiques suivants :

### 2-RESSOURCES-INPUTS, reçues par le SYSTEME

sur lesquelles s'exerce l'activité du SYSTEME en vue de réaliser les résultats (OUTPUT)

Exemple : Bon de commande pour un système "Réception et contacts-clients".

3-RESSOURCES-PROPRES, utilisées (ou modifiées) par le SYSTEME qui permet la réalisation du processus de transformation d'INPUT en OUTPUT sans qu'elles-mêmes ne deviennent partie intégrante des OUTPUTS et donc sans qu'elles ne quittent le système.

Exemple : le "fichier"-clients pour un système "gestion des ventes".

le même fichier pour le système "édition des factures" par ex.

### 4-RESSOURCES-OUTPUTS, produites par le SYSTEME

qui résulte du traitement des OUTPUTS (et des ressources propres), elles concrétisent la réalisation de la FONCTION.

Exemple : La facture pour les systèmes "Facturation-clients" et "édition des factures"...

Dans un parallélisme simpliste, on comparerait ressources-inputs, - propres et - outputs aux termes familiers de MATERIAUX, MATERIELS et PRODUITS FABRIQUES.

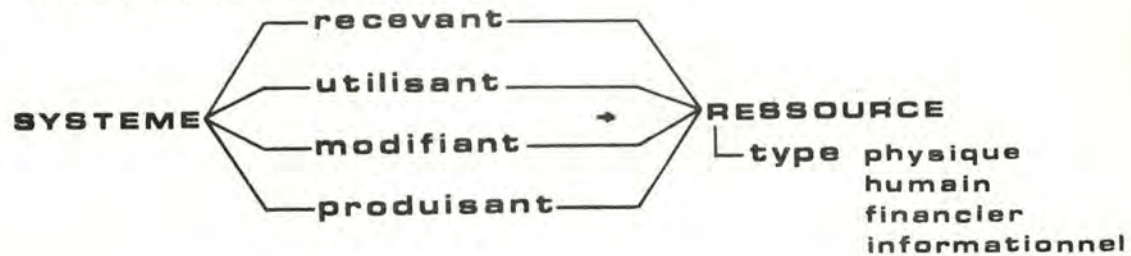
En ce qui concerne les ressources de type informationnel, on ne peut manquer de signaler que

d'une part, nous n'avons fait aucune hypothèse ou restriction sur le degré de décomposition d'une telle ressource; aussi une facture et une de ses rubriques élémentaires sont, dans notre esprit, 2 exemples de ressources informationnelles.

D'autre part, le terme de ressource-PROPRE recouvrira souvent celui, moins déroutant, de "donnée mémorisée" (ou history).



Schématiquement, le fonctionnement statique du système d'informations peut se ramener à



Le cinquième élément caractéristique de tout SYSTEME, propre à la STRUCTURATION DES TRAITEMENTS, est sa

**5- Décomposabilité en (sous)-SYSTEMES**

ou la faculté de fractionner le processus de conversion d'INPUTS en OUTPUTS

Exemple : les (sous-) systèmes Réception, Magasin, Livraison, Facturation et gestion crédit-clientèle du système Administration des ventes; ou encore les systèmes traitement-bon-entrée-en-stock, traitement-bon-commande et traitement-fin-fichier du système traitement-du-stock.

Tout en maintenant le terme unique de SYSTEME quelque soit le niveau de décomposition, nous avons jugé bon de pouvoir caractériser un SYSTEME, par la spécification d'un TYPE, en fonction précisément de ce niveau et admettre ainsi les 3 adjectifs suivants : ELEMENTAIRE, COMPOSITE ou GROUPE.

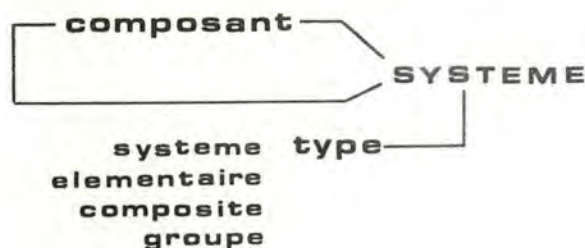
Dans notre esprit, COMPOSITE et GROUPE ne sont pas synonymes mais s'appliquent à deux situations (moments) différentes rencontrées lors de toute analyse d'un système d'informations.

Comme nous l'avons évoqué précédemment, toute analyse passe par des phases de DECOMPOSITION et de SYNTHÈSE ou restructuration. A ce niveau de STRUCTURATION DES TRAITEMENTS, notre système de documentation autorisera donc la description

- de SYSTEMES COMPOSITES progressivement décomposés en
- SYSTEMES ELEMENTAIRES, eux-mêmes éventuellement progressivement regroupés (SYNTHÈSE) au sein de
- SYSTEMES GROUPES.(°)

La souplesse du système de documentation impose qu'on laisse le choix des règles de décomposition ou de synthèse aux utilisateurs de ce système.

La structuration des traitements donne schématiquement ceci



(°) On comprendra que nous n'envisageons ici que des regroupements ayant un sens du point de vue de la conception (mêmes ressources manipulées, même événement-déclencheur - Cfr. infra) à l'exclusion des regroupements effectués lors des phases de construction physique ou des facteurs de performance et d'optimisation interviendront.



Les éléments caractéristiques - spécialement de 1 à 4 - que nous venons d'énumérer et de commenter, décrivent l'aspect STATIQUE du fonctionnement d'un système d'informations. Pour compléter et parfaire cette approche, il convient de préciser la manière dont chaque élément intervient dans le système et évolue dans le temps.

Le dernier élément caractéristique de tout SYSTEME insiste donc sur l'aspect DYNAMIQUE de son fonctionnement sous le vocable, par exemple, de

#### 6- Conditions dynamiques de mise en oeuvre

- qui donnent des indications de fréquence et/ou de volume d'intervention d'une ressource dans le système
- ou qui signalent le(s) événement(s) qui déclenche(nt) la mise en oeuvre d'un SYSTEME (en particulier de TYPE processus).<sup>(°)</sup>

Exemple : le système "édition de rappels" est à activer lorsque le solde-client est en découvert de plus de 5.000 F" et "depuis plus de 2 mois".

PROCESSUS, entité dynamique : synonyme de SYSTEME

A partir de ce qui précède, on constate donc qu'un SYSTEME peut être "vu" non seulement comme une combinaison de ressources, mais aussi comme une entité dynamique. Aussi avons-nous introduit la notion de PROCESSUS qui, à nos yeux, doit être considérée comme synonyme de celle de SYSTEME, mais permet d'insister davantage sur l'aspect dynamique de celui-ci.

A ce propos, vu la grande similitude entre le fonctionnement d'un système d'informations et celui des systèmes d'exploitation, il est commode d'emprunter à la théorie des operating system la notion - et sa définition - de PROCESSUS. En effet, Crocus spécifie explicitement :

Un processus représente l'exécution d'un programme comportant des instructions et des données. C'est une entité dynamique.

(Crocus, 1975).

Le processus répond donc à notre notion CONVENUE de SYSTEME (une fonction, des ressources - inputs, - propres, - outputs...) mais s'y ajoute une caractéristique dynamique plus prononcée qui le fait naître lors du déclenchement de l'exécution d'une PROCEDURE et mourir à la fin de cette exécution. Le processus est donc une mise en oeuvre !

la **procédure**, ressource-propre particulière, est donc, dans cette optique une description STATIQUE de l'ensemble ordonné des règles qui doivent être observées et des actes qui doivent être accomplis pour parvenir à la production des OUTPUTS lors de sa mise en oeuvre. Ceci complète donc le schéma du fonctionnement statique de la façon suivante :

**SYSTEME**

de ——— **procédure: instruction**  
                                  **enonce** ———

(°) Cette notion d'événement est approfondie dans la suite du texte.



**REMARQUE:**

La structure d'une procédure et le contenu d'une instruction gagneraient à être examinés et étudiés plus sérieusement dans le sens d'une plus grande formalisation (Cfr. le "langage des conception" de REMORA) mais en veillant à garder des expressions naturelles, proches du langage courant pour décrire ces procédures.

Faute de temps, le choix de la structure d'une procédure et du contenu d'une instruction sera laissé à l'analyste et notre système de documentation les traitera comme du texte libre (commentaire) sans contrôles, syntaxiques ou autres.

Nous venons d'insister sur l'aspect dynamique d'un SYSTEME (ou d'un PROCESSUS). Il est assez logique d'envisager maintenant par quel mécanisme la mise en oeuvre d'un PROCESSUS peut être assurée.

Nous allons tenter de brièvement expliciter cette notion d'EVENEMENT-déclencheur, peut-être plus familière aux responsables des "Centres d'exploitation" (planning des travaux,...) et aux habitués des operating systems (synchronisation des processus...) qu'elle ne l'est aux concepteurs ou analystes.

**311.2****EVENEMENT (\*)**

En effet, qui dit mise en oeuvre d'un processus à un moment donné, dit mécanisme déclencheur, c'est-à-dire un mécanisme permettant l'activation ou la désactivation (blocage) d'un processus en fonction de ce qui se passe dans l'organisation.

Un événement informe le système d'informations de ce qui se passe dans son système-objet (organisation), il correspond toujours à une opération de création ou de clôture...

(Golvers, 1976).

Soulignons, de nouveau, le parallélisme avec les systèmes d'exploitation et ses mécanismes de synchronisation - indirectes - de processus :

Le problème de la synchronisation consiste à construire un mécanisme qui permette à un processus  
d'en bloquer un autre (...)  
d'activer un autre processus (...)  
Dans un mécanisme d'activation indirecte, la synchronisation met en jeu un ou plusieurs objets intermédiaires (notamment des EVENEMENTS) connus des processus coopérants...

(Crocus, 1975).

---

(\*) Une étude plus complète de cette question et de la notion générale d'évènement est souhaitable surtout si certains contrôles de cohérence du fonctionnement dynamique du système sont prévus au niveau de l'outil .



Remarquons, également, qu'une fois le système d'informations en exploitation opérationnelle, il peut se faire qu'un processus soit déjà actif lorsqu'un événement, devant entraîner son activation, intervient. Il faut donc laisser à l'utilisateur du système de documentation la possibilité de définir de façon précise l'effet de l'opération d'activation lorsqu'on se trouve dans cette circonstance, sous forme d'un commentaire, par exemple, en attendant une formalisation plus grande de cette notion. (°)

Evènement simple et composé.

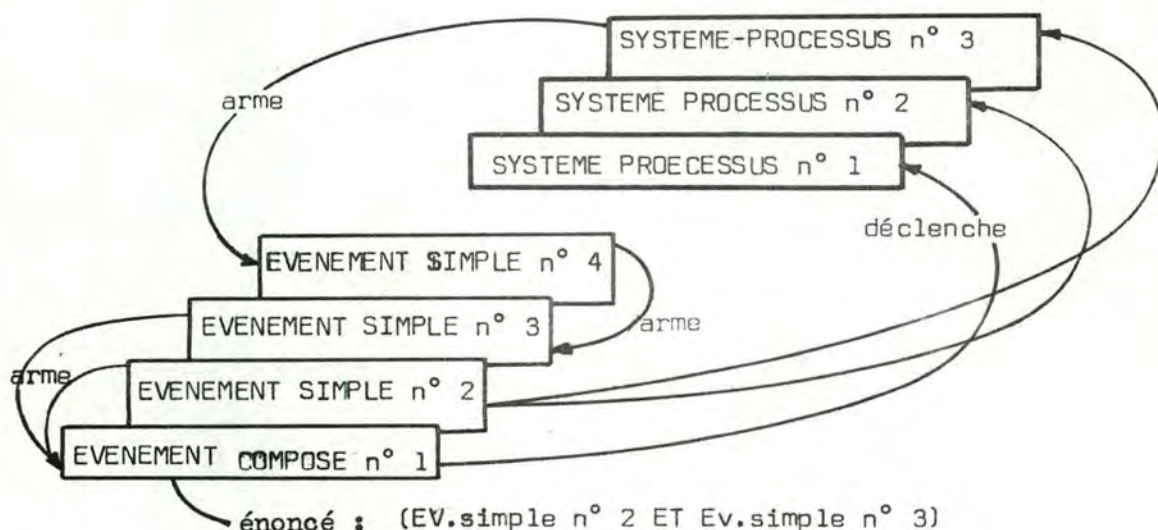
Nous avons, jusqu'à présent, laissé supposer qu'un seul évènement pouvait être rattaché à un processus comme, par exemple,

- la survenance d'un fait externe (occasionnel, accidentel ou périodique)
- la fin - ou une instruction particulière - d'un processus qui arme ainsi un évènement (déclenche ainsi un évènement)
- la surveillance d'un fait interne au système d'informations (niveau d'une variable par rapport à un seuil...)

Toutefois, on peut très bien imaginer que l'activation ou la désactivation d'un processus soit associée non plus à la survenance d'un seul EVENEMENT (simple), mais à l'occurrence "conjointe de 2 évènements ou à la survenance de l'un ou l'autre des 2, ou plus généralement à une expression booléenne (EVENEMENT COMPOSE) d'EVENEMENTS SIMPLES.

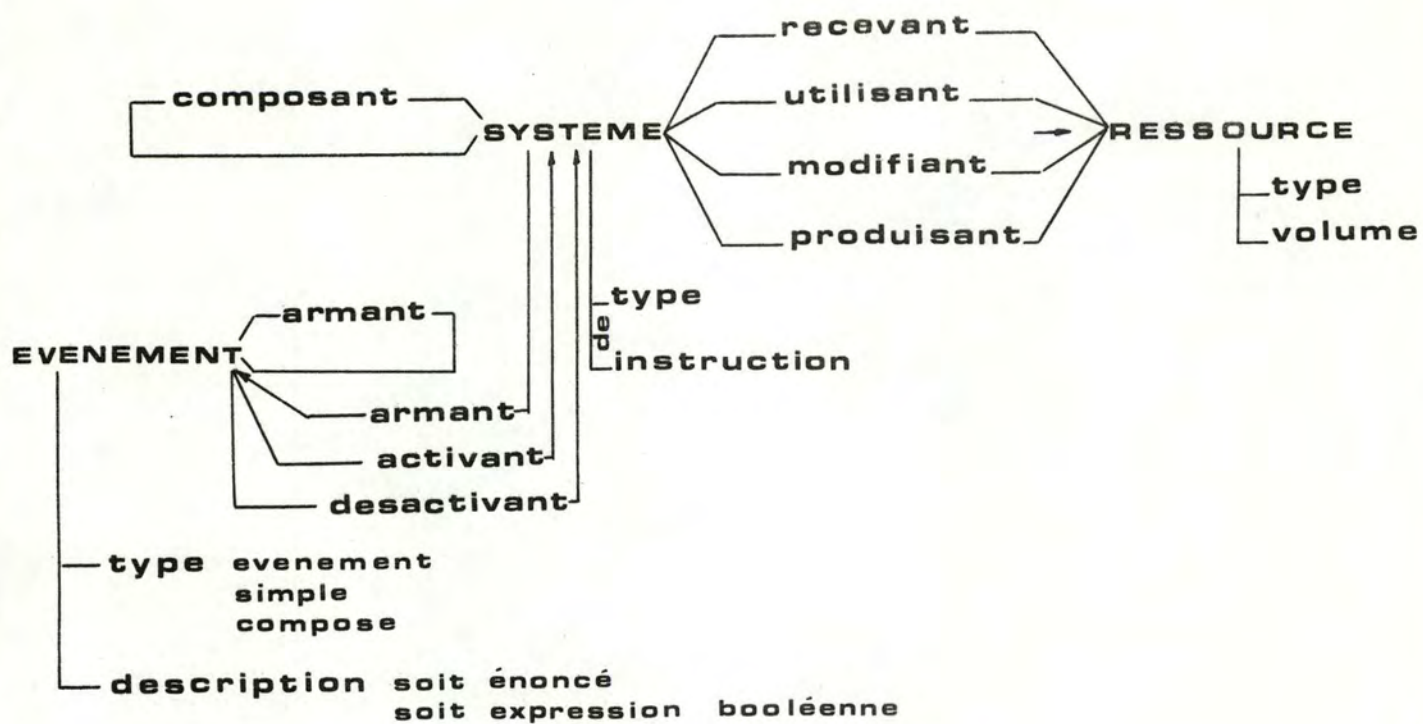
Par ailleurs, la survenance d'un EVENEMENT peut non seulement activer des processus mais également armer d'autres événements (déclencher...)

Un exemple graphique fera mieux comprendre les possibilités qu'autorise notre "modèle"



Le schéma suivant visualise et récapitule la partie de la terminologie ayant trait au fonctionnement statique et dynamique du système et à la structuration des traitements :







**STRUCTURATION et REPRESENTATION des DONNEES**

Comme nous l'avons rappelé dans la première partie (Pôles de description d'un système d'informations), le fonctionnement n'est qu'un volet de la description d'un système d'informations: il reste maintenant

non seulement à présenter les concepts qui permettront de structurer ou de relier les données entre elles (puisque nous avons déjà réglé le problème de la structuration des traitements)

mais aussi à établir la JONCTION entre ces deux aspects de la description : le fonctionnement et la structuration.

Car, contrairement à la plupart des systèmes de documentations rencontrés, nous avons préféré la solution - fort proche de celle retenue par CASCADE - qui consiste à gérer deux concepts (RESSOURCE INFORMATIONNELLE et INFORMATION) selon que la même entité est vue

- une fois comme élément (ressource) du modèle de fonctionnement
- une autre fois comme élément du modèle de structuration

Nous tenterons, bien entendu, de justifier, dans la suite de cet exposé, cette distinction apparemment subtile.

Notre terminologie (ou modèle) propose donc trois niveaux de structuration des données et les concepts correspondants :

- le niveau de la PERCEPTION de la réalité en termes d'ENTITES (OBJET ou ASSOCIATION) offrant toutes les garanties souhaitées de stabilité.
- le niveau de l'INFORMATION à propos de ces OBJETS, de ces ASSOCIATIONS et de leurs propriétés; Par ailleurs, ce sont les liens, entre ces "INFORMATIONS", qui permettront de décrire la STRUCTURE des RESSOURCES INFORMATIONNELLES manipulées par les traitements.
- le niveau de la REPRESENTATION de ces "INFORMATIONS" au moyen de "DONNEES" - conceptuelles - ELEMENTAIRES.

Ce schéma logique à trois niveaux correspond, en fait, à ce que prônent certains concepteurs de bases de données comme, par exemple :

- |   |  |
|---|--|
| La théorie infologique des bases de données distingue |  |
| - le système-objet                                    |  |
| - les informations à propos de ce système-objet       |  |
| - la représentation                                   |  |
| - (la mémorisation et l'accès aux informations)       |  |

(Sundgrend, 1974).



## 312.1

## ENTITES de l'ORGANISATION REELLE

Le fonctionnement d'un système d'informations est généralement sujet à des modifications assez fréquentes dues à l'évolution de l'organisation et aux exigences des utilisateurs.

Il est, par contre, apparu qu'une description - globale - de l'organisation qui ne ferait appel qu'à des OBJETS, des ASSOCIATIONS d'OBJETS et leurs propriétés, indépendamment de leur utilisation dans un tel ou tel traitement, témoignerait d'une beaucoup plus grande STABILITE et pourrait, par conséquence, être pris comme élément de référence. (°)

Définissons brièvement ces trois concepts dont l'intérêt est précisément d'être d'usage courant :

Un OBJET est une entité abstraite ou concrète qui est vue par l'utilisateur comme un tout ayant une existence propre et à propos de laquelle on veut rassembler de l'information.  
(Un client, une commande, un produit...).

Tandis qu'une ASSOCIATION est une mise en relation de plusieurs objets ou chacun d'eux joue un rôle particulier et à ce titre l'existence d'une association dépend de l'existence des objets qu'elle relie.  
(Une ligne de commande est une association reliant les objets : commande et produit...).

Enfin, les OBJETS et les ASSOCIATIONS sont perçues via un ensemble de PROPRIETES qui sont des caractéristiques que les utilisateurs attribuent à ces objets ou les associations. Ces PROPRIETES peuvent être reconnues, attribuées ou créées par l'organisation.

## REMARQUE IMPORTANTE.

Pour appréhender l'organisation ou la réalité, les utilisateurs ont pris l'habitude de considérer des types ou des classes d'objets et d'associations et donc de propriétés; (Exemple : les personnes qui achètent chez moi ont un ensemble de caractéristiques communes et, à ce titre, sont appelés "CLIENTS"...).

Lors de l'analyse et, a fortiori, au niveau d'un système de documentation, c'est davantage le nom (Existence) de ces types ou classes qui sont pris en compte. Aussi, est-ce par un certain abus de langage que nous parlerons d'un OBJET ou d'une ASSOCIATION au lieu, respectivement, d'une classe d'OBJETS et d'une classe d'ASSOCIATIONS; réservant le terme d'OCCURENCE (contenu) pour l'élément d'une classe.

---

(°) Pour une justification de ces affirmations, voir notamment  
(Benci, Bodart, Bogaert, Cabanes, 1976)  
(Sundgren, 1974 et 1976).

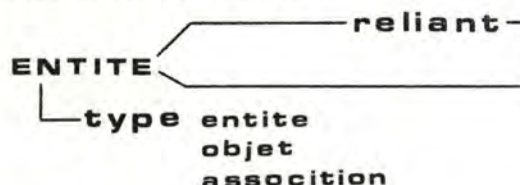


ENTITE : OBJET OU ASSOCIATION.

Dans notre modèle, le terme générique d'ENTITE recouvrira aussi bien la notion d'OBJET que celle d'ASSOCIATION, considérées donc comme 2 types possibles d'ENTITES.

Par ailleurs, la relation "RELIANT" permettra d'indiquer - ou d'enregistrer - les (ENTITES de type) OBJETS qui jouent un rôle dans une (ENTITE de type) ASSOCIATION.

Schématiquement, on vient donc de mettre en évidence une nouvelle classe d'objets dans notre modèle :



QUANTIFICATION ET INSERTION DANS LE TEMPS. (°)

Afin de mieux cerner une ENTITE, nous avons donné la possibilité aux utilisateurs de ce système de documentation, d'enregistrer des renseignements complémentaires - qui prendront toute leur signification lors des phases de REALISATION (Choix de l'organisation physique : accès et mémorisation) - sous forme d'une indication de taille (volume) et d'un indice de stabilité de l'ENTITE.

En effet, dans la mesure où nous nous intéressons à des classes d'éléments, il peut être utile et judicieux d'ajouter des renseignements quantitatifs sur le volume des occurrences d'un OBJET

ou d'une ASSOCIATION (à telle occurrence d'un OBJET correspond au minimum et au maximum tel nombre d'occurrences de l'autre OBJET relié au premier par l'ASSOCIATION).

ou comme le signalent deux auteurs :

Ainsi les tailles de classes d'objets doivent être renseignées comme caractéristiques importantes.

(Langefors, 1974).

On veillera également à mentionner la cardinalité des liaisons binaires entre objets.

(Bodart, 1976).

Par ailleurs, il sera souvent intéressant de situer les ENTITES sur une "échelle de stabilité" en fonction

- de l'évolution du nombre d'occurrences d'une ENTITE
- des modifications plus ou moins fréquentes des valeurs d'une occurrence.

Ainsi, par exemple :

(°) Le problème de quantification et d'insertion dans le temps des objets et des associations mériterait à lui seul un travail de modélisation assez considérable (études statistiques de population et de sa stabilité, études des mouve-.../...



Il faut renseigner si une classe est "invariante",  
"variant lentement" ou "volatile".  
(Langefors, 1974).

A titre d'hypothèse de travail on peut distinguer cinq  
grands types de "relations" qui décriront la structure  
fonctionnelle du point de vue de leur degré de variabilité.  
(Bodart, Bogaert, 1975).

Les "lots" peuvent être classés selon trois points de vue:  
d'état, d'observation et procédural; chaque type donnant  
lieu à deux natures : permanente ou situationnelle, évé-  
nementielle ou historique, interrogative ou conservation-  
nelle.  
(Thuly, Saunier, 1976).

Nous avons retenu une échelle à 4 composantes - du plus stable  
au plus volatile :

- HISTORIQUE, par ex. l'association "passé-client" (client, produit)
- PERMANENT, par ex. l'objet "client"
- DE-SITUATION, par ex. l'association "compte-client" (client, agence).
- TEMPORAIRE, par ex. l'objet "bon de commande".

Rien n'empêche, cependant, l'analyste d'en choisir une autre puisque  
cette caractéristique (INDICE de STABILITE) est traitée sous forme  
de texte libre (comme la plupart des "attributs" de notre modèle). (°)

Exemple

Objet : CLIENT

(propriétés : le fait d'avoir un numéro, un nom, une  
adresse...)

(indice de stabilité : PERMANENT)

COMMANDE

(propriétés : le fait d'avoir un numéro et d'être passée  
à une certaine date...)

(indice de stabilité : TEMPORAIRE)

Association : ORDRE reliant les objets : CLIENT et COMMANDE

(cardinalité : (1-n) (1-1) signifiant que cette association  
relie un client à au moins un et au plus n commandes tandis  
qu'à une commande ne correspond qu'1 et 1 seul client).

## 312.2

### INFORMATION : concept de jonction

La même remarque, faite à propos du concept de SYSTEME est de rigueur  
pour celui d'"INFORMATION" : le terme d'INFORMATION, stricto sensu,  
est à distinguer mentalement des autres utilisations générales du même  
terme évoquées par ailleurs.

Nous tenterons également de montrer que ce terme - en particulier dans  
sa version élémentaire - opère la jonction entre le fonctionnement du  
système et la structuration des données (ainsi que leur représentation).

.../... (°) ments de cette population : fréquence, date et période de  
création ou de destruction, et ceci sans négliger l'aspect  
probabiliste des choses.

On comprendra dès lors que, faute d'une telle étude, ce que  
nous proposons (en particulier à l'échelle de variance) n'



De plus, suivant le niveau de décomposition ou d'agrégation, nous distinguerons plusieurs types d'INFORMATION: composite, élémentaire et groupée.

Sa forme la plus dépouillée et la plus fondamentalement utile au dessein du système de documentation est donc appelée :

#### INFORMATION ELEMENTAIRE : DEFINITION

Elle est très précisément définie comme étant  
la DESIGNATION (apportant une "connaissance" à propos) d'  
une seule propriété (au sens, bien entendu, de classe de  
propriétés) d'une seule ENTITE (toujours au sens de classe)  
de type - OBJET ou ASSOCIATION.

Par ailleurs, une occurrence d'INFORMATION (de type) ELEMENTAIRE devra être localisée

- dans le temps par une date d'observation ou une période de validité
- dans l'espace par l'indication d'un lieu défini.

Cette notion est donc à rapprocher de celle d'"e-concept" de Langefors et de Sundgren ou de celle d'"information élémentaire" de Bodart :

Une information élémentaire définit la configuration de données élémentaires qui désigne une propriété d'usage au sein de l'organisation associée à un objet au cours d'une période de temps ou à une date donnée et en un lieu défini.

(Bodart, 1976).

L'e-concept est l'ensemble des e-messages qui informent à propos des objets d'une même classe...

Un e-message "informe" à propos d'une propriété associée à un objet identifié pendant une période donnée.

(Langefors, 1974).

Le schéma de notre modèle s'enrichit donc de la façon suivante :



#### INFORMATION ET RESSOURCE INFORMATIONNELLE.

La définition précédente ne présente, effectivement, qu'un des aspects du concept d'INFORMATION car celui-ci est également en relation avec le concept de RESSOURCE INFORMATIONNELLE; Deux concepts, dont on aurait pu penser, dans une vision rapide, qu'ils étaient redondants.

Mais il faut rappeler que, conventionnellement, une RESSOURCE INFORMATIONNELLE ne présupposait en aucune façon une quelconque idée de "niveau de décomposition" et recouvrait aussi bien la notion de fichier ou d'ensemble d'informations que celles d'enregistrement et de rubrique élémentaire.



En fait, c'est précisément par l'intermédiaire du concept d'INFORMATION que l'on va pouvoir décrire la structure de décomposition ou d'agrégation d'une RESSOURCE INFORMATIONNELLE.

Cette volonté de dissociation - au niveau même des concepts - entre les RESSOURCES et leur description, se justifie notamment dans la mesure où il est relativement fréquent de trouver des RESSOURCES INFORMATIONNELLES différentes qui peuvent être décrites par les mêmes ensembles de composantes. Que l'on songe, par exemple, ici à l'image d'un fichier des clients et à celle du "fichier-cartes" de mise à jour de ce même fichier-clients. Cette observation met donc en évidence l'existence de STRUCTURES de décomposition ou d'agrégation indépendantes du contenu d'une information ou d'un ensemble d'informations (vus alors comme ressources de processus déterminés de traitement).

La finesse de la décomposition sera obtenue par l'existence de différents types d'INFORMATION : nous avons déjà présenté sa forme la plus ELEMENTAIRE, il ne nous reste plus qu'à expliquer les INFORMATIONS de type COMPOSITE OU GROUPE.

Dans la pratique donc, l'INFORMATION ELEMENTAIRE résulte d'une étape de décomposition ou de désagrégation que l'analyste applique au départ d'une INFORMATION plus globale et plus brute, plus directement accessible à l'observation. Ce concept plus général, et plus "temporaire" d'ailleurs, sera appelé :

- INFORMATION de type COMPOSITE.

Celle-ci est donc supposé être un conglomerat d'INFORMATIONS ELEMENTAIRES ou éventuellement d'autres INFORMATIONS COMPOSITES. A ce titre, elle désigne donc une ou plusieurs propriétés d'une ou plusieurs ENTITES et elle peut décrire une RESSOURCE plus ou moins agrégée.

Donc, au fur et à mesure que se poursuit la phase de décomposition, chaque INFORMATION COMPOSITE est progressivement DECOMPOSEE, éventuellement en autres INFORMATIONS COMPOSITES décrivant des ressources plus désagrégées, mais finalement en INFORMATIONS ELEMENTAIRES, (autant de niveaux de décomposition qu'on le désire).

Enfin, quand la phase de décomposition est achevée, lorsque toutes les INFORMATIONS COMPOSITES ont été remplacées par des INFORMATIONS ELEMENTAIRES - et qu'éventuellement la structure des ENTITES a complètement été établie - on peut envisager des regroupements - volontaires maintenant - d'INFORMATIONS ELEMENTAIRES, au sein :

- d'INFORMATIONS de type GROUPE.

Ce type d'INFORMATIONS désigne alors plusieurs propriétés (éventuellement toutes) mais associées à une seule ENTITE (toujours au sens de classe, bien entendu). Cette notion est donc à comparer à celle de e-concept ou d'unité d'informations (°) :

---

(°) A notre connaissance, seuls Langefors et ses "disciples" (Sundgren, CASCADE, CADIS...) font clairement la distinction, dans les groupements d'information élémentaires, entre les phases de décomposition et celles de synthèse ou d'agrégation.

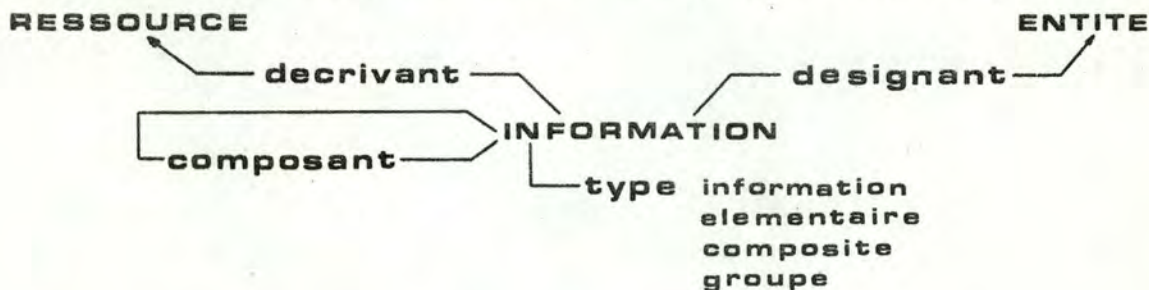


Le concept consolidé est un ensemble de e-concepts, comprenant donc plus d'une propriété à propos d'une classe d'objets...  
(Langefors, 1974).

Nous appellerons unité d'informations, ce groupement des propriétés d'usage d'un même objet utilisées par un traitement homogène.

(Bodart, 1976).

Nous pouvons donc compléter notre schéma de la façon suivante :



#### AVANTAGES DE DISSOCIER INFORMATION ET RESSOURCE INFORMATIONNELLE.

Le fait de localiser au même endroit, via le concept d'INFORMATION de notre modèle, la description de la Structure de la RESSOURCE (et non pas son contenu) permet d'éviter la répétition inutile de cette description pour des RESSOURCES (INFORMATIONNELLES), analogues ou identiques, intervenant dans un SYSTEME donné ou même plusieurs SYSTEMES.

Reprenons l'exemple, évoqué précédemment, du fichier-permanent des clients et du fichier-mouvement des mises à jour (ou de création) de ce même fichier-clients.

De plus, de par notre définition elle-même, l'INFORMATION ELEMENTAIRE est unique - et localisée à un seul endroit - dans l'ensemble du Système de documentation, indépendamment de son intervention ou de sa mise en oeuvre dans plusieurs structures (COMPOSITION D'INFORMATIONS) et donc de plusieurs RESSOURCES INFORMATIONNELLES.

Cette caractéristique d'unicité est notamment significative si l'on pense à l'élaboration nécessaire du vocabulaire de gestion commun à l'organisation (dictionnaire des informations élémentaires. Elle est aussi à rapprocher de la préoccupation des travaux de LESCA (Grenoble) : "Approche des Systèmes d'Informations par le vocabulaire de gestion des entreprises. (°)

Il n'est peut-être pas inutile d'évoquer ici une remarque de cet auteur ayant trait à la signification quantitative d'un tel dictionnaire :

Le nombre des descripteurs (-informations élémentaires) du vocabulaire de gestion de l'entreprise est le plus souvent compris entre un plancher de 400 et un plafond qui est de 4.000.  
(Lesca, 1977).

(°) In INFORMATIQUE et GESTION, n° 85, mars 1977 (Dossier).



Par ailleurs, cette caractéristique d'unicité constitue aussi une forme d'autodiscipline garantissant l'homogénéité ou l'intégrité du système élaboré. Alors que l'analyste dans le cadre de son étude de "fonctionnement" interne d'un SYSTÈME dispose d'une certaine souplesse, c'est-à-dire d'une certaine liberté de modification; il ne dispose plus des mêmes degrés de liberté pour modifier la structure des RESSOURCES elles-mêmes. Il risque, en effet, d'atteindre d'autres SYSTEMES mettant en oeuvre des RESSOURCES à la même structure, à la même description en terme d'INFORMATIONS.

Une RESSOURCE est rattachée à un SYSTÈME (éventuellement subdivisé), tandis qu'une INFORMATION est globale pour tout le système d'informations !

Enfin, on comprendra aisément que, pour la même raison, le problème des "synonymes et des polysèmes" perd de son acuité. Car ce n'est plus par un effort - personnel - intellectuel qu'une éventuelle relation de synonymie ou de polysémie (au niveau de l'INFORMATION ELEMENTAIRE) est mise en évidence mais bien par l'existence - de facto - de liaisons entre des RESSOURCES et des INFORMATIONS :

(°)

- Plusieurs RESSOURCES au nom éventuellement différent, **rattachées** à des SYSTEMES différents mais reliées à une seule INFORMATION ELEMENTAIRE
- Plusieurs RESSOURCES au nom **identique** mais à la "portée" fixée par l'appartenance à des SYSTEMES différents et reliées à des INFORMATIONS ELEMENTAIRES différentes (ayant donc un sens différent puisqu'il s'agira d'une autre propriété de la même ou d'une autre ENTITE).

Ces différents avantages doivent être mis en balance avec l'inconvénient d'une apparente redondance au niveau des termes utilisés !

Notre modèle offre, par ailleurs, la possibilité à l'analyste de compléter la définition d'une INFORMATION par une caractéristique supplémentaire grâce à l'introduction d'un nouveau concept : la FORMULE (objet-attribut d'une INFORMATION).

#### la formule

Ce concept est destiné à recouvrir toutes les expressions que chaque occurrence d'une INFORMATION ELEMENTAIRE doit vérifier pour être valide. Il peut s'agir

- d'une règle d'obtention - éventuellement conditionnelle - de cette INFORMATION ELEMENTAIRE à par tir d'autres INFORMATIONS ELEMENTAIRES  
Ex. TOTAL = TOTAL-HORS-TAXE \* TVA; pour l'INFOELEM (TOTAL)
- d'une condition à satisfaire (existence d'une valeur, par exemple :)  
Si SEXE = 'FEMININ' et si ETAT-CIVIL = 'CELIBATAIRE' alors NOM-JEUNE-FILLE = -; pour l'INFOELEM (NOM-JEUNE-FILLE)
- ou d'une quelconque contrainte à respecter.

---

(°) Mais le problème des synonymes et des polysèmes subsiste toujours, bien sûr ! Seule la façon de l'aborder présente à notre avis quelques avantages.



Tout comme pour les INSTRUCTIONS des procédures, le même effort de formalisation reste à faire, nous nous contenterons d'insister sur la similitude de nature d'une INSTRUCTION et de celle d'une FORMULE et donc sur la nécessité qu'il y a à développer et d'utiliser le même formalisme dans les deux situations.

## la protection des données

Nous allons une fois de plus faire état du parallélisme entre les operating systems et les systèmes d'informations, car les problèmes de protection de données sont sensiblement identiques dans les deux cas. Voici comment Crocus pose le problème pour un système d'exploitation, mais, en fait, cela s'applique également à un système d'informations.

La coexistence, à l'intérieur d'un système, d'informations appartenant à différents utilisateurs impose la protection de ces informations contre les erreurs (...) Par exemple, un utilisateur peut souhaiter n'autoriser la consultation ou la modification de ses informations privées qu'à certains utilisateurs explicitement spécifiés; plusieurs utilisateurs peuvent aussi avoir des droits différents sur une même ressource.

(Crocus, 1977).

Nous avons donc - pour un système d'informations -

- un ensemble d'informations (ressources) à protéger, dont l'utilisation doit être contrôlée.
- Un ensemble d'utilisateurs, par ressource informationnelle, dont il convient de privilégier le propriétaire au sens - un peu restrictif peut-être - de (Thuly, Saunier, 1976) : "le premier utilisateur qui ait éprouvé le besoin de l'information est responsable de celle-ci et notamment de sa mise à jour, mais il peut céder certains de ses DROITS à d'autres utilisateurs."
- un ensemble de droits par utilisateur (mode d'accès : consultation, modification, création ou suppression).

D'autres auteurs ajoutent une 4ème classe d'éléments, en particulier :

- Un ensemble de conditions ou de prédicats devant être vérifiés pour réaliser l'accès...

(Benci, Bodart, Bogaert, Cabanes, 1976).

de la même façon qu'en operating system, on admet que l'on puisse "calculer le mode d'accès par une fonction dépendante de l'utilisateur et de la ressource".

Nous avons résolu le problème de la façon suivante :

1. Toute INFORMATION pourra avoir un PROPRIETAIRE (responsable de la définition de l'information)
2. Au niveau des RESSOURCES (correspondant à cette INFORMATION), nous avons adopté la solution dite de la liste d'accès (sous forme d'un nouvel objet-attribut), c'est-à-dire qu'à chaque RESSOURCE on pourra associer une liste des utilisateurs autorisés (à avoir accès à cette information) éventuellement accompagnés de leurs droits.

On complète donc notre schéma du modèle de la façon suivante.





## 312.3

**DONNEE : élément de REPRESENTATION**

Comme nous l'avons répété tout au long de ce travail, le champ d'investigations de celui-ci - et donc, de notre modèle - est limité aux activités de CONCEPTION des systèmes d'informations. Les concepts et leurs relations (en particulier les modes de représentation et d'accès des informations) ayant trait aux phases de REALISATION ne seront pas envisagés. Nous nous contenterons de préparer le travail en considérant les DONNEES ELEMENTAIRES - conceptuelles - qui constituent une INFORMATION ELEMENTAIRE (OU GROUPEE).

Pour mieux faire comprendre la différence entre une donnée (physiquement enregistrée) et une DONNEE conceptuelle, nous empruntons à (Langefors, Sundgren, 1975) la comparaison suivante :

Dans le langage courant, un mot ne prend sa totale signification que dans son utilisation dans une phrase. Mais pour ce faire, c'est que le mot a, bien entendu, un sens - sa définition au dictionnaire - pour celui qui entend la phrase. Le sens de ce mot correspond donc à une idée auquel ce mot (physique = ensemble de sons) est associé.

Ex. Les mots physiques "Chiens", "Dog" ou "Hund" sont représentatifs du même concept (idée). Quand nous employons le mot "chien" nous n'avons apporté aucune information à propos de la réalité en dépit du fait que tout le monde comprend le sens d'un tel mot.

Ainsi, de la même façon, une DONNEE ELEMENTAIRE-conceptuelle (= différente d'une donnée physique, élément de représentation des informations) - ne possède pas de signification en soi. Ce n'est en fait qu'un ensemble de valeurs de même nature (elle a donc un sens). Exemple : âge, prix, ...

Une INFORMATION ELEMENTAIRE - ou éventuellement GROUPEE - pourra donc être vue comme une combinaison de DONNEES conceptuelles.

Par ailleurs, de par la définition même d'une INFORMATION ELEMENTAIRE (OU GROUPEE), nous pouvons distinguer 3 types de DONNEES pour tenir compte de la structure d'une telle INFORMATION :

les DONNEES IDENTIFIANT l'ENTITE concernée par la propriété  
 LOCALISANT - dans le temps et l'espace- cette propriété  
 CARACTERISANT la ou les propriété(s) désignée(s) par  
 l'INFORMATION.

Cette structure est notamment mise en évidence par les deux auteurs suivants :

Les données élémentaires associées à une information élémentaire seront réparties en 3 groupes :  
 Celles qui identifient l'objet  
 Celles qui désignent la propriété d'usage de l'objet (...)  
 Celles qui localisent dans le temps et dans l'espace cette propriété.

(Bodart, 1976).



Il semble qu'un e-message contienne toujours 3 "références"  
 ou "termes conceptuels"  
 Quel objet ou ensemble d'objets est visé  
 quelle(s) propriété(s)  
 à quel moment.

(Langefors, 1974).

Une telle définition des DONNEES ELEMENTAIRES (= ensemble de valeurs) a incité certains auteurs à considérer qu'une INFORMATION (en particulier ELEMENTAIRE) pouvait être représentée sous la forme d'une relation (un peu particulière) dont les domaines seraient précisément les DONNEES ELEMENTAIRES et à tirer profit de cet outil mathématique (°). Rien n'empêche dans notre terminologie ou modèle d'avoir cette vision d'une INFORMATION même si a priori nous ne l'avons pas spécialement privilégiée.

Ex. d'information élémentaire PRIX du PRODUIT :  
 "représentée" par la configuration de données élémentaires  
 identifiantes : N° de PRODUIT, NOM de PRODUIT  
 localisantes : DATE, PERIODE, SUCCURSALE  
 caractérisantes : PRIX.

Dans la mesure où une DONNEE ELEMENTAIRE n'est qu'un ensemble de valeurs (sans signification en soi), il est normal que notre modèle permette la définition de cet ensemble, en fonction de la conception du problème étudié. Et cette définition devra pouvoir se faire

- soit par l'énumération des valeurs possibles (éventuellement sous forme d'une table de codification)
- soit par la définition des frontières de l'ensemble
- soit par une fonction de calcul.

Ce sera le rôle de l'objet-attribut (de DONNEE) **valeur** !

Deux autres attributs permettront de compléter les renseignements à propos d'une DONNEE, il s'agira du FORMAT de celle-ci (étendue et nature de la donnée). Ex. 8 caractères alphanumériques.  
 et de l'éventuelle UNITE de MESURE des valeurs d'une DONNEE (en particulier CARACTERISANTE) représentative de l'unité dimensionnelle de la propriété désignée par l'INFORMATION à laquelle est rattachée la DONNEE.

#### IDENTIFICATION D'UNE ENTITE.-

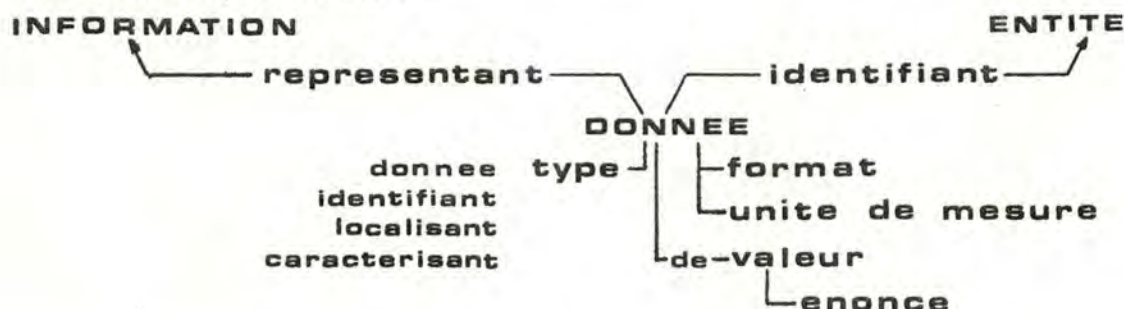
Nous avons également ajouté pour l'analyste la possibilité de relier directement toute occurrence d'un OBJET ou d'une ASSOCIATION à sa ou ses DONNEE(S) IDENTIFIANTE(S).

Ex. l'ASSOCIATION reliant les OBJETS clients, produits, départements, pourra être directement reliée à ses IDENTIFIANTS :  
 numéro de client, numéro de produit et nom de département.

(°) Voir notamment à ce propos (Benci, Bodart, Bogaert, Cabanes, 1976).



La "représentation des informations" est donc pris en considération de la façon suivante



31.3

### IDENTIFICATION des OBJETS de notre MODELE

Chaque occurrence d'un objet de notre modèle (SYSTEME, EVENEMENT, RESSOURCE, ENTITE, INFORMATION, DONNEE) est identifié par une RACINE - ou identifiant court, par ex, de 8 caractères- et par un TYPE ou indication d'appartenance à une des 6 classes citées ci-dessus ou même à une sous-classe

Exemple: l'information élémentaire Prix du Produit est parfaitement identifiée de la façon suivante:

RACINE : PRIPROD

TYPE : INFORMATION - ELEMENTAIRE

De plus, la définition d'une occurrence pourra être complétée et précisée par une DESCRIPTION -plus longue, par ex, d'une centaine de caractères, traitée comme du texte libre par le système.

Mais tandis que pour les 5 objets SYSTEME, EVENEMENT, ENTITE, INFORMATION et DONNEE il nous semble souhaitable d'imposer qu'une RACINE et un TYPE identifient complètement une seule occurrence d'objet, pour le dernier objet (RESSOURCE), nous avons estimé plus souple de pouvoir qualifier une occurrence d'un tel objet par celle du SYSTEME qui manipule cette ressource.

Exemple: la ressource "prix du produit" reçue-par le système "facturation" :

RACINE : PXPRO

TYPE : RESSOURCE INFORMATIONNELLE

identifiant : PXPRO DE FACTUR, RESSINFO

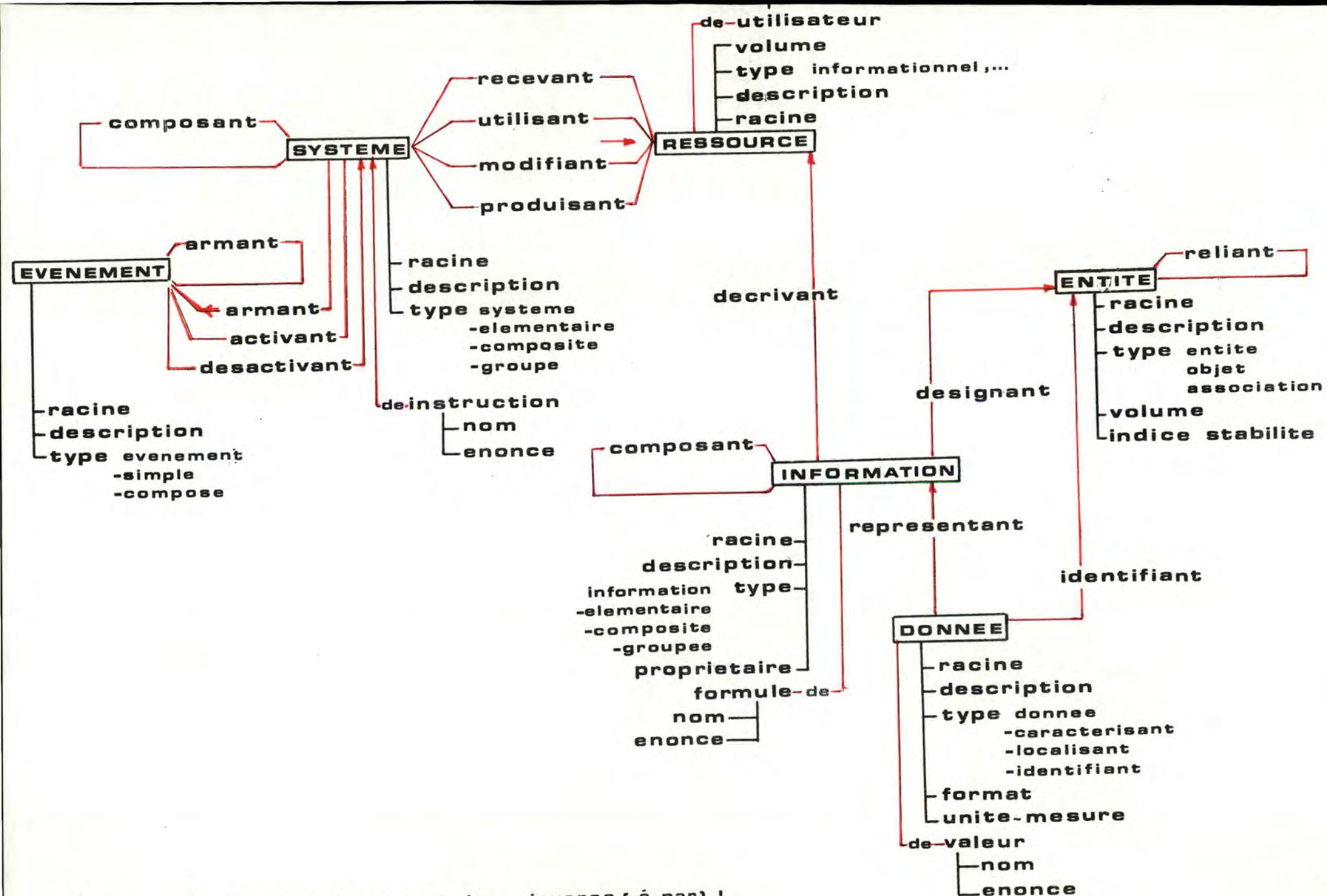
Enfin, pour les 4 objets-attributs d'un objet principal INSTRUCTION, FORMULE, UTILISATEUR et VALEUR nous avons adopté le même genre de solution. Outre un NOM (équivalent de la RACINE) et un TYPE, il faut qu'ils soient qualifiés par l'occurrence de l'objet principal dont ils ne sont que des attributs.

Exemple: le processus élémentaire TRFACTUR et les 5 instructions de sa procédure:

NOM : 1	ENONCE (équivalent à la DESCRIPTION) : ...
NOM : 2	ENONCE : ...
NOM : 3	ENONCE : ...
NOM : 4	ENONCE : ...
NOM : 5	ENONCE : ...

De plus, le fait d'avoir donné un NOM (ou un numéro) à chaque INSTRUCTION (FORMULE, UTILISATEUR ou VALEUR) permet une plus grande souplesse de modification puisque l'on peut ainsi ne modifier, par ex, qu'une instruction et non toute la procédure d'un processus. (cfr chapitre suivant : LANGAGE PROPOSE)





toutes les relations (ant) ont leur inverse (-é-par) !



## B.2 LANGAGE PROPOSE.

Dans l'étude des systèmes de documentation existants, nous avons répertorié plusieurs classes de langages possibles pour manipuler les concepts de la terminologie convenue et donc pour établir (enregistrer) de nouveaux "faits" (=renseignements) dans la base documentaire :

- descripteurs sous forme tabulaire (MINOS, REMORA)
- langages dits naturels reposant sur une phraséologie particulière (ISDOS)
- langage de commande de type "manipulation de bases de données" (DD, CASCADE...) en batch ou en interactif.

On pourrait également imaginer un langage sous forme de grille-écran et ses programmes interactifs (CADIS, DIFO,...).

Le choix entre ces différentes solutions est loin d'être évident !

Deux remarques s'imposent :

Primo, on pourrait très bien imaginer que plusieurs langages différents soient d'application pour un même modèle et donc un même stock documentaire.

Secundo, rien n'empêche, momentanément, d'utiliser le langage de requêtes du SGBD utilisé pour gérer le stock documentaire, en attendant d'installer (compilateur,...) un langage plus adapté aux problèmes de l'analyse.

Mais quels sont les objectifs d'un tel langage (critère de choix) ? Le premier objectif est assurément de se rapprocher le plus possible du langage courant ou en tout cas de la façon de penser de l'analyste, tout en restant suffisamment formalisé que pour être reconnaissable et exploitable par un automate.

Le second objectif - nous avons insisté sur celui-ci par ailleurs - est la souplesse d'utilisation du système qui laisse la plus grande liberté possible à l'analyste dans l'organisation de son travail de documentation.

Nous avons estimé qu'un langage de type "manipulation de bases de données" un peu particulier et dédié à notre modèle répondrait - notamment et dans une certaine mesure à ces deux objectifs.

Enfin, la forme adoptée pour la présentation et l'explication des instructions ne doit pas laisser croire que notre objectif était de proposer la "grammaire" complète du langage proposé. Nous avons simplement voulu présenter dans ses grandes lignes un langage que, faute de temps, nous n'avons pas pu développer plus profondément. Le formalisme adopté nous a simplement aidé dans notre tâche par la concision et le haut pouvoir explicatif de ses expressions.

Pour terminer, nous espérons que l'exemple d'utilisation fourni permettra de mieux "visualiser" notre terminologie en même temps que notre langage.



## CREATION

ou établissement d'un nouveau "fait" dans le stock documentaire : nouvelle **occurrence** d'un objet, d'un attribut, d'un objet et d'une relation avec une autre **occurrence** ou, enfin, d'une relation.

<commande> = CREER ou CR

1. L'instruction suivante a pour effet de créer une ou plusieurs **occurrences** de l'objet dont on mentionne le TYPE dans la partie gauche ( par rapport au ":" ) de l'instruction. La partie droite est toujours réservée à l'identification du nouveau "fait" à introduire dans le stock.

```

<instruction A> := <expression 1> : <liste de couples>;
<expression 1> := <commande> <type>

<liste de couples> := <couple d'identification> | <liste de couples>,
                        <couple d'identification>
<couple d'identification> := <valeur de RACINE> * <valeur de RACINE>
                        ( <valeur de DESCRIPTION> )
<type> := SYSTEME | SYSELEM | SYSCOMPT | SYSGROUP | (°)
        EVENEM | EVENSIMP | EVENCOMP |
        RESSINFO | RESSHUM | RESSPHYS | RESSFINFO |
        ENTITE | OBJET | ASSOCIAT |
        INFORM | INFOLEM | INFOCOMP | INFOGROUP |
        DONNEE | DONIDENT | DON_LOCAL | DONCARACT

```

Ex. CREER ASSOCIAT : LIGCOM (LIGNE-DE-COMMANDE);  
 CR EVENCOMP :EV44 (EV2 et EV5 ou EV17);  
 CR EVENSIMP :EV17 (CHAQUE FIN DE MOIS), EV2 (SOLDE CLIENT DEBITEUR),  
 EV5;

2. L'instruction suivante permet de créer une **occurrence** d'un attribut ou d'un objet-attribut d'une occurrence d'un des objets principaux (Cfr 1,-) du modèle.

```

<instruction B> := <expression 2> : <valeur de caractéristique>;
<expression 2> := <commande> <caractéristique d'objet> DE <type>
                ( <liste d'identifiants> )
<Caractéristique d'objet> := <attribut> | <objet-attribut>
<Attribut> := RACINE | DESCRIPTION | TYPE | (pour les objets)
              NOM | ENONCE | (pour les objets-attributs)
              VOLUME | (pour ressinfo ou entrée) INDICSTABIL |
              PROPRIO | (pour entité)
              FORMAT | UNITEMES (pour donnée)
<Objet-attribut> := INSTRUCT | FORMULE | UTILISAT | VALEUR
<Liste d'identifiants> := <identifiant> | <liste d'identifiants>,
                        <identifiant>
<Identifiant> := <valeur de RACINE> |
                <valeur de RACINE> DE <valeur de RACINE>

(Ce dernier cas, si l'on souhaite qualifier une RESSOURCE
 par le SYSTEME qui la manipule)

<Valeur de caractéristique> := <valeur d'ATTRIBUT> | <liste de paires>
<Liste de paires> := <paire> | <liste de paires>, <paire>
<Paire> := <valeur de NOM> | <valeur de NOM> ( <valeur d'ENONCE> )

```

(°) ou PROCESSUS | PROCELEM | PROCCOMP | PROCGROUP |







## 32.2 SUPPRESSION

d'une occurrence d'objet, d'attribut, de relation, de relation et du premier terme (occurrence d'objet) de celle-ci.

Une fois la précision suivante établie :

**<commande> := CR I SUP**

tout ce que nous avons établi pour la commande de CREATION est automatiquement applicable à la situation présente de SUPPRESSION.

Ex. SUP SYSTEME : RAPCLI (RAPPEL CLIENT); (1)  
 SUP INSTRUCT DE PROCELEM (FACTOR) : 5; (2)  
 ou SUP INSTRUCT DE PROCELEM (FACTOR) : 5 (SOLDE=SOLDE + PU \* QUANTIT) (2)  
 SUP RESSINFO RECU-PAR PROCELEM (FACTOR) : SOLDINIT;  
 SUP RELATION RESSINFO DECRIT-PAR INFOCOMPOS(ENTETE): IENTGEN;

Cette dernière instruction ne supprime que l'occurrence de la relation; sans effacer les 2 occurrences d'objets qui étaient reliées par la Relation.

## 32.3 MODIFICATION

Nous avons décidé que toute modification porterait exclusivement sur une occurrence d'attribut ou d'objet-attribut.

Ainsi dans le cas où

**<commande> := MOD**

seule l'instruction B aura, à nos yeux, un sens :

**<instruction B> := <expression 2> : <valeur de caractéristique>;**

valeur de caractéristique indique ici la ou les nouvelle(s) valeur(s) qui doit(en)t donc remplacer celle(s) qui existai(en)t précédemment.

Ex. MOD TYPE DE OBJET (CLICLI) : CLIENT ; (dorénavant l'objet CLICLI s'appellera CLIENT)  
 MOD UNITEMES DE DONCARACT (PRIX DE IPXPROD) : MILLIERS FRANCS;  
 (Avant c'était par exemple des CENTAINES FRANCS)  
 MOD INSTRUCT DE PROCELEM (FACTOR) : 5 (nouvel énoncé de l'instruction numéro 5);  
 MOD VALEUR DE DONCARACT (BAREME) : 10 (98 = ASSISTANT SOCIAL OU ASSIMILE); alors que précédemment c'était (98 = ASSISTANT SOCIAL OU INFIRMIERE) par exemple.

## 32.4 OBTENTION

d'un renseignement contenu dans la base documentaire. Il pourra s'agir de

- toutes les occurrences d'un objet (un TYPE donné)
- l'occurrence d'un attribut ou des occurrences d'un objet-attribut
- l'occurrence de l'objet rattaché à un autre (connu) par une relation (connue)

La structure des instructions est semblable aux 3 premières instructions de CREATION :



<commande> := OBT(énir)

1. L'interrogation suivante aura pour résultat l'obtention par le système (sous forme d'image-écran ou de listing, par exemple) de toutes les occurrences d'un objet (TYPE) donné.

<interrogation A> := <expression 1> ::

Remarque : En plus de la RACINE et de la DESCRIPTION d'une occurrence, le système devra fournir le TYPE précis de celle-ci, car nous admettons qu'un TYPE (dans l'interrogation) soit le "nom générique" d'un objet (SYSTEME par ex.). Dans cet exemple, le système précisera s'il s'agit d'un PROCELEM, d'un PROCCOMP ou d'un PROCGROUP.

Ex. : OBT INFOELEM ::

le système restituera par cette instruction le dictionnaire des informations élémentaires de l'organisation.

2. L'interrogation suivante permettra d'obtenir la valeur d'un attribut de l'occurrence de l'objet cité ou toutes les occurrences (nom et énoncé) d'un objet-attribut.

<interrogation B> := <expression 2> ::

Ex. OBT INDICSTABIL DE ENTITE (COMPLI) :: (réponse : "DE SITUATION")

OBT DESCRIPTION DE SYSTEME (FACT) :: (réponse : "SERVICE RECEPTION  
COMMANDES ET FACTURATION...")

OBT INSTRUCT DE PROCCOMP (FACTOR) :: (réponse : 1(énoncé instruction),  
2( idem ),  
3( idem ),  
Etc.

3. Le système restituera la ou les occurrences d'objet dont on connaît la liaison avec une autre occurrence d'objet par l'interrogation suivante :

<interrogation C> := <expression 3> ::

La remarque faite dans le paragraphe 2 est également d'application dans le cas présent.

Ex. : OBT RESSINFO DECRIT-PAR INFOELEM (INUMCLI) ::

La réponse mettra en évidence toutes les utilisations (ressources) du "numéro de client" et fera apparaître les éventuels synonymes de celui-ci.

OBT EVENEM ACTIVANT PROCGROUP (TRESOR) ::

(Réponse : le ou les événements-déclencheur (racine, description et type) de l'exécution du processus "Traitement-gestion-trésorerie").

Enfin, nous avons jugé bon d'ajouter une 4ème classe d'instructions :



4. L'interrogation suivante a pour résultat de produire tous les descendants ou tous les ascendants d'une occurrence d'objets (SYSTEME OU INFORMATION) reliés par une relation de COMPOSITION (<RELATION> := COMPOSANT COMPOSE-PAR ).

<interrogation 4> := <expression 4> ::

<expression 4> := <commande> STRUCTURE <type> <liste de liaisons>

Ex.: OBT STRUCTURE INFORM COMPOSANT INFOCOMP (IBONCDE) ::

Le système restituera les descendants (les autres INFORMATIONS) de l'information composite "BON-DE-COMMANDE-CLIENT" tels qu'ils ont été enregistrés précédemment (avec, par ex. une indication de niveau - de décomposition)

1. IBONCDE, BON DE COMMANDE CLIENT, INFOCOMP

2. IPARTCLI, INFORMATION CLIENT, INFOCOMP

3. INOCLI, NUMERO, INFOELEM

3. INOMCLI, NOM, INFOELEM

3. IADRCLI, ADRESSE, INFOELEM

2. IPARTPRO, INFORMATION A PROPOS DES PRODUITS CDES, INFOCOMP

3. INUMPRO, NUMERO, INFOELEM

3. IQUANT, QUANTITE COMMANDEE, INFOELEM

2. ITRANSP, INFORMATION SUR MOYENS TRANSPORT, INFOCOMP

Etc.

Sans cette instruction, on aurait pu arriver au même résultat mais avec un nombre plus élevé d'instruction (interrogation C)

Une première instruction pour obtenir les niveaux 2.

et par niveau 2, une instruction pour obtenir les niveaux 3,

et ainsi de suite jusqu'au niveau des INFORMATIONS ELEMENTAIRES

(ou des PROCESSUS ELEMENTAIRES). (°)

## 32.5

### COMPLEMENT : MACRO

Toujours avec le souci de réduire le texte sans nuire à la clarté des expressions, on pourrait compléter ce langage par la possibilité de définir des MACROS (CASCADE)

<définition d'une MACRO> := DEFINIR MACRO <identification de la macro>

,

' instructions du langage avec des  
' paramètres formels

,

,

,

FIN <identification de la macro> ;

<exécution d'une macro> := EXECUTER MACRO <identification de la macro>  
( <liste des paramètres actuels> );

(°) Il aurait peut-être été également avantageux d'étendre le champ d'application de cette instruction aux EVENEMENTS et à la relation définie sur eux (ARMANT, ARME-PAR). Il semble toutefois qu'il y ait rarement un nombre élevé de niveaux de décomposition d'un EVENEMENT et que par conséquent cette modification ne se justifie pas.



Le langage ainsi défini devrait pouvoir rendre des services à l'analyste dans ses activités de documentation et d'analyse. Il resterait à construire un compilateur et à tester, dans la réalité, ses possibilités mais aussi la qualité principale que nous lui voulions : être suffisamment naturel et le moins ésotérique possible.

Par ailleurs, nous ne l'avons pas envisagé, dans un premier temps, mais, en plus du langage, il ne serait peut-être pas superflu de prévoir la possibilité pour l'analyste d'obtenir des rapports -standard ou non- à la demande.



## 3.3

**EXEMPLE D'UTILISATION.**

L'exemple qui va suivre a pour unique objectif l'illustration de ce qui précède. Aussi n'a-t-il nullement la prétention d'être exhaustif !

Cette application pratique du langage et de la terminologie atteindrait donc son objectif, si elle contribuait à donner une meilleure idée du fonctionnement de ce système de documentation.

Imaginons, au sein d'une cellule de l'organisation (administration-ventes), un processus de traitement (exécution des commandes) dont la fonction serait, à partir d'un bon de commande (client), de contrôler la solvabilité du client  
examiner si la commande est exécutable à partir des stocks  
mettre à jour les fichiers d'informations concernés  
éditer la facture.

Supposons, que les données manipulées se limitent, en entrée, aux BONS DE COMMANDES ayant le profil suivant

INUMERO DE CDS ! NUMERO DE CLIENT ! NUMERO DE PRODUIT ! QUANTITE COMMANDEE !

en sortie, aux FACTURES reprenant les rubriques suivantes :

INUMERO DE FACTURE !  
INUMERO DE CLIENT ! NOM DE CLIENT !  
INUMERO DE PRODUIT ! QUANTITE LIVREE ! PRIX UNITAIRE !  
! TOTAL HORS TAXE ! TVA, TOTAL !

en données mémorisées, à quelques indication sur les clients :

INUMERO DE CLIENT ! NOM DE CLIENT ! SOLDE ! QUANTITE CUMULEE !

les produits :

INUMERO DE PRODUIT ! QUANTITE EN STOCK ! PRIX UNITAIRE !

Nous avons sélectionné quelques instructions du langage pour tâcher de donner une bonne idée de la description de cette application selon les 3 approches de fonctionnement, de structuration et de représentation.

- Commençons par quelques instructions de "structuration des traitements" décrivant

- le rattachement de ce processus à un système plus vaste :

```
CR(EER) PROCESCOMP COMPOSANT SYSTEME (ADMIENTE) : EXECDE (TRAITEMENT-EXECUTION-COMMANDES);
```

- la décomposition de ce processus composite en processus élémentaire :

```
CR PROCESELEM COMPOSANT PROCESCOMP(EXECDE): SAISCDE (TRAITEMENT -SAISIE-COMMANDES),  
FINCDE(TRAITEMENT-FIN-FICHER), SOLDBT (TRAITEMENT-SOLDE-DEBITEUR-CLIENT),  
QUINSUF (TRAITEMENT-QUANTITE-INSUFFISANTE), EDITFAC (TRAITEMENT-EDITION-FACTURE);
```



- Au niveau du "fonctionnement statique" de ce processus, examinons les ressources informationnelles dont il a besoin ou qu'il produit :

CR RESINFO RECU-PAR PROCESCOMP (EXECDE) : BCDE (BON-DE-COMMANDE),  
QUCDEE (QUANTITE-COMMANDEE), NCU (NUMERO-DE-CLIENT), NPRO (NUMERO  
DE-PRODUIT);

CR RESINFO PRODUIT-PAR PROCESCOMP (EXECDE): FACT (FACTURE);

CR RESINFO UTILISE-PAR PROCESCOMP (EXECDE) : FCLI (FICHIER-CLIENTS),  
FPRO (FICHIER-PRODUITS), PRUNIT (PRIX-UNITAIRE-PRODUIT), NUMCLI  
(NUMERO-DE-CLIENT), NUMPRO;

CR RESINFO MODIFIE-PAR PROCESCOMP (EXECDE) : SOLCLI (SOLDE-CLIENT),  
QUSTOC (QUANTITE-EN-STOCK), QUCUMUL (QUANTITE-CUMULEE-CLIENT); (...)

Après avoir passé en revue les ressources - informationnelles -  
INPUT, OUTPUT ou mémorisée, décrivons la ressource particulière,  
PROCEDURE de ce processus (traitement - exécution-des-commandes) :

CR INSTRUCTION DE PROCESCOMP (EXECDE) :  
1 (SAISCDE),  
2 (SI FIN DE FICHIER ALORS FINCDE),  
3 (SI SOLCLI 0 ALORS SOLDBT),  
4 (SI QUSTOCK QUCDEE ALORS QUINSUF),  
5 (QUSTOC = QUSTOC - QUCDEE),  
6 (QUCUMUL = QUCUMUL + QUCDEE),  
7 (SOLCLI = SOLCLI + QUCDEE \* PRUNIT),  
8 (EDITFAC);

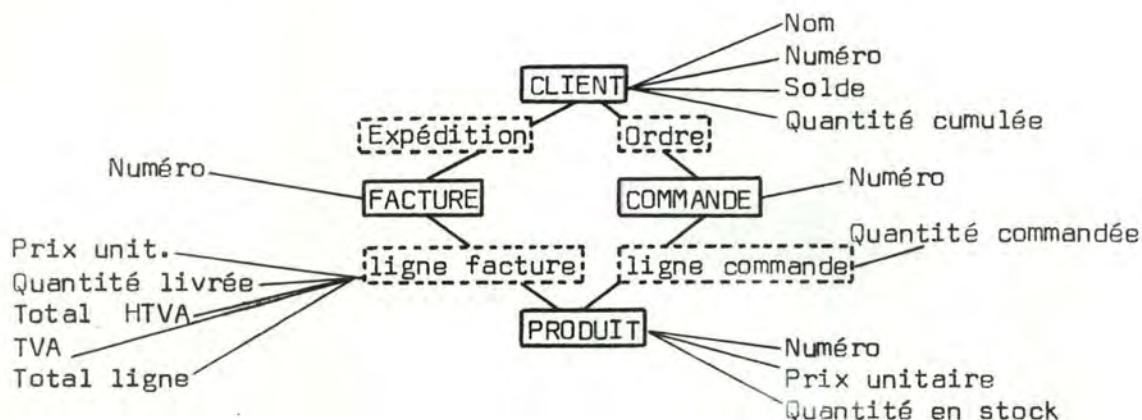
- Complétons cette approche "fonctionnement" du processus par quelques indications sur l'aspect "dynamique" de celui-ci :

CR EVENSIMP ACTIVANT PROCESCOMP (EXECDE) : EVCDE 1 (RECEPTION-D4UN-  
BON-COMMANDE);

CR VOLUME DE RESINFO (BCDE) : 100-200/JOUR ;

CR VOLUME DE RESINFO (FCLI) : 4500; (...)

- Nous avons retenu + objets et + associations pour "structurer les données"-de notre -mini-application (de façon stable), dont voici une présentation graphique :





Il nous faut donc traduire ce schéma en instructions de notre langage; et, tout d'abord pour décrire les objets et les associations de cette structure :

```
CR OBJET : CLIENT, FACTURE, PRODUIT, COMMANDE ;
CR INDICSTABIL DE OBJET (CLIENT) : PERMANENT;
CR INDICSTABIL DE OBJET (COMMANDE) : TEMPORAIRE ;
(..)
CR ASSOCIAT RELIANT OBJET (FACTURE, PRODUIT) : LIGFAC (LIGNE-DE-FACTURE);
CR ASSOCIAT RELIANT OBJET (COMMANDE,PRODUIT) : LIGCOM (LIGNE-DE-COMMANDE);
CR VOLUME DE ASSOCIAT (LIGFAC) : 1 - 10 - 1 - 1 ;
(...)
```

On aurait pu procéder autrement pour enregistrer une association.

Par exemple :

```
CR ASSOCIAT : LIGCOM (LIGNE-DE-COMMANDE);
CR RELATION OBJET RELIE-PAR ASSOCIAT (LIGCOM) : COMMANDE, PRODUIT;
(...)
```

Continuons le travail de structuration

- en détaillant les "informations" à propos des propriétés des objet ou des relations répertoriés sur le schéma précédent (DESIGNATION)
- tout en les reliant aux ressources - informationnelles - qu'elles décrivent (DESCRIPTION)
- pour dégager - ou situer dans - la structure de ces ressources (COMPOSITION).

```
CR INFOCOMP DECRIVANT RESINFO (BCDE) : IBCDE;
CR INFOCOMP DECRIVANT RESINFO (FCLI) : IFCLI;
CR INFOCOMP DECRIVANT RESINFO (FPRO) : IFPRO;

CR INFOELEM COMPOSANT INFOCOMP (IBCDE, IFCLI), DECRIVANT RESINFO (NCLI,
NUMCLI), DESIGNANT OBJET (CLIENT) : INUMCLI (NUMERO-DE-CLIENT-PROPRE);
```

Remarquons la présence, de facto, de synonymes (NCLI, NUMCLI) pour l'entité "numéro de client", sans instruction particulière (clause SYNONYM ou autre) rendu possible par cette séparation RESSOURCE (propre à des SYSTEMES désignés) et INFORMATION (global pour tous les SYSTEMES).

```
CR INFOELEM COMPOSANT INFOCOMP (IBCDE), DECRIVANT RESINFO (QUCDEE),
DESIGNANT ASSOCIAT (LIGCDE) : IQUCDEE (QUANTITE-COMMANDEE);
CR INFOELEM COMPOSANT INFOCOMP (IBCDE, IFPRO), DECRIVANT RESINFO
(NPRO, NUMPRO), DESIGNANT OBJET (PRODUIT) : INUMPRO (NUMERO-DE-
PRODUIT);
CR INFOELEM COMPOSANT INFOCOMP (IFPRO), DECRIVANT (PRUNIT), DESIGNANT
OBJET (PRODUIT): IPRUNIT (PRIX-UNITAIRE-DU-PRODUIT);
CR INFOELEM COMPOSANT INFOCOMP(IFPRO), DESIGNANT OBJET (PRODUIT),
DECRIVANT (QUSTOC) : IQUSTOK (QUANTITE-EN-STOCK-DE-CE-PRODUIT);
```

Quelques derniers exemples :



CR RELATION INFOELEM DESIGNANT ASSOCIAT (LIGFAC) : ITVA (TAXE-VALEUR-AJOUTEE);  
 CR FORMULE DE INFOELEM (ITOTAL) : EQ1 (ITOTAL = ITOTHS \* ITVA); (...)

o Terminons par quelques indications de "Représentation" pour l'INFORMATION (PRIX DU PRODUIT)

CR DONIDENT REPRESENTANT INFOELEM (IPRUNIT) : NOPRO, NOPRO;  
 CR DONLOCAL REPRESENTANT INFOELEM (IPRUNIT) : DATE (DATE-DEFINITION-PRIX-PAR-DIRECTION);  
 CR DON CARACT REPRESENTANT INFOELEM (IPRUNIT) : PRIX;  
 CR FORMAT DE DONCARACT (PRIX) : 8 NUM;  
 CR UNITMES DE DONCARACT(PRIX) : FRANCS;  
 CR VALEUR DE DONCARACT( PRIX):  
 1(125), 2(175), 3(250), 4(500);

Enfin, il serait peut-être utile de relier les occurrences d'un objet ou d'une association à leur(s) identifiant(s). Pour l'objet "produit" cela pourrait se faire par l'instruction suivante :

CR RELATION DONIDENT IDENTIFIANT OBJET(PRODUIT) : NOPRO;



## CONCLUSIONS

---

Il ne paraît pas utile, sous prétexte de conclusion, de reprendre en d'autres termes le contenu de la table des matières. Mais il peut être opportun de rappeler certaines options que nous avons adoptées.

C'est tout d'abord l'affirmation de l'indépendance souhaitable d'un outil de documentation à l'égard des méthodes d'analyse.

C'est aussi le souci omniprésent d'un système "naturel" accessible et proche de l'utilisateur, évitant notamment l'excès de concepts et de conventions et en proposant un langage dont la syntaxe soit le plus proche de la formulation courante.

C'est ensuite la distinction proposée entre "ressource informationnelle, (modèle de fonctionnement) et "information" (modèle de structuration) destinée à rendre compte de la distinction observée entre contenu et structure de l'information perçue.

Enfin, le travail retient - de recherches actuelles sur les bases de données - certaines options pour l'organisation et la représentation des informations au sein d'une "base de données" sous forme notamment de la trilogie "objet - association - propriété" et de leur représentation conventionnelle au niveau des "données élémentaires".

Il reste cependant deux importantes remarques à formuler :

D'une part, le présent travail ne doit pas être considéré comme un aboutissement puisqu'il a laissé plus d'un problème non - totalement - résolu et nécessiterait, à ce titre, un prolongement sous forme d'une étude dépassant l'envergure d'un mémoire. Cette étude devrait notamment se faire - et pour commencer - dans le sens

- d'une formalisation plus grande au niveau du modèle
- d'un effort de précision accrue du langage et de son compilateur
- d'une meilleure connaissance des outils à mettre en place (besoins réels, objectifs, limites, moyens...)
- mais surtout d'une confrontation de ce système - si possible installé - avec la réalité des organisations et de leurs applications à informatiser.

D'autre part, notre stage nous a permis de constater qu'un système de documentation, indépendamment de sa valeur sur le plan théorique, n'a de sens que s'il est employé par les utilisateurs (analystes et autres) soit parce qu'imposé par une direction consciente du problème, soit - et c'est d'autant mieux - parce que les utilisateurs en perçoivent le bien fondé. Il faut à ce propos faire remarquer que la plupart des systèmes retenus (2ème partie) ont une origine



universitaire (si l'on veut bien excepter Minos, Data Dictionary, PAC...). Porter donc un jugement sur l'opérationnalité et l'utilisation actuelle et effective des systèmes de documentation exigerait d'une part une étude plus exhaustive des systèmes existants (et notamment commercialisés) et d'autre part une enquête - sur le terrain - auprès des utilisateurs, effectifs et potentiels.



## REMERCIEMENTS

Nous remercions toutes les personnes qui nous ont aidé dans l'élaboration du présent travail et notamment

- F. BODART, Directeur de l'Institut d'Informatique, FNDP
- H. BOGAERT, Assistant, FNDP
- R. LESUISSE, Assistant, FNDP
- J. THULY, Directeur du Service "Méthodes et Système" de la Société AUTOMOBILES PEUGEOT à PARIS.



## BIBLIOGRAPHIE CONSULTÉE

### Articles de base :

- TEICHROEW, RATAJ, HERSHEY; an introduction to computer-aided documentation of user requirements for computer - based information processing systems; in GROCHLA, SZYPERSKY; Information Systems and organisation structure; Berlin, 1975; pg 438-464.
- LANGEFORS; Information systems; in Information Processing 1974 IFIP; North-Holland Co, 1974; pg 937-943.

### Ouvrages ou articles de référence :

- BENCI, BODART, BO GAERT, CABANES; Modelling in data base management systems; in Proceedings of the IFIP working conference 1976 North Holland, reprint by Nijssen, Brussels.
- BODART, BOGAERT; Objectifs associés à la structure fonctionnelle; Namur 1975 (Notes internes FNDP).
- BODART; Introduction à l'analyse fonctionnelle des systèmes informatiques de gestion; cours FNDP, 1976, Namur.
- CABANES; Banques de Données; Cours FNDP et CNAM, 1976-1977.
- CROCUS; Systèmes d'exploitation des ordinateurs; Dunod, n° 686, 1975.
- LEMOIGNE; Les Systèmes d'informations dans les organisations; PUF 1973.
- NADLER; La conception des systèmes; VANDER 1973
- SUNDGREN; Using the infological approach in the design of data base; in Systemeering 75; Lundeborg; Studentlitteratur Lund 1976.
- SUNDGREN; Conceptual foundations of the infological approach to data base; in Data base management, KLIMBIE et KOFFEMAN; North Holland Publishing Co 1974; pg 61, 95.
- THULY, SAUNIER; Objectifs et conceptions d'une base de données, Ed. Informatique; Paris 1976.

### Systèmes existants :

#### Généralités ou synthèses.

- COUGER; Evolution of business System analyses techniques; in COUGER, KNAPP; System Analysis techniques; Wiley 1974.
- ; Dossier "Méthodes nouvelles de conception de systèmes d'informations"; in INformatique et Gestion, n° 85, mars 1977.  
     en particulier-ROLLAND, Un système de pilotage de la conception des systèmes d'information;
- PECCOUD, GIRAUDIN; Réflexions sur la construction d'un prototype pour faciliter la conception d'une application informatique;



- TEICHROEW, A survey of languages for stating requirements for computer-based systems, in AFIPS Conference proceedings, FJCC, 1972, pg 1201-1223.
- TEICHROEW; Problem Statement language in MIS, in COUGER, KNAPP; System Analysis techniques, Wiley 1974, pg 310, 327.
- TAGGART; Developping an organisation's information inventory, in Management informatics, 1974; vol 3, n° 6, pg 223-292.

#### Ads.-

- LYNCH, ADS, A technique in Systems documentation, in Data Base, Vol 1, n° 1, Printemps 1969.
- LESUISSE; Un système de documentation fonctionnelle : ADS; référence incomplète.
- NUNAMAKER; Computer-aided analysis and design of information systems, in Communication of ACM, Vol 19, n° 12, Décembre 1976.

#### Tag.-

- IBM; The Time automated Grid Systems, in COUGER, KNAPP, System Analysis techniques, Wiley 1974.

#### Minos.-

- ; Package Minos IIIA, guide de l'utilisateur et guide de l'exploitation; documents internes Automobile Peugeot, 1977.

#### Remora.-

- ROLLAND; La base des objets élémentaires de Remora et étude du langage de conception; documents internes, Université de Nancy, 1975.

#### Data Dictionary.-

- PLAGMAN, ALTSHULER; A DD/DS within the contexte of an integrated corporate data base; in AFIPS, FJCC, 1972, Vol 2; pg 1133 et suiv.
- IBM; Data dictionary information manual; documentation constructeur, 1976.

#### Cascade, Cadis et IA2.-

- SOLVBERG, SKYLSTAD, AANSTAD; Cascade - A computer based documentation system; in Comput Information Systems Analysis and design by LANGEFORS, BUKENKO; Student Litteratur Lund 1973.
- SOLVBERG, TORLEN, ANGLAEND; Proposals to Standard Flowcharts in System design; in Systemeering 75 by LUNDEBERG; Studentlitteratur Lund 1976.
- BUKENKO, KALLHAMAR; Cadis : Computer-aided design of information system; in Computer-aided information systems analysis and design by LANGEFORS, BUKENKO; Studentlitteratur Lund 1973.
- LUNDEBERG; Information Analysis (IA2) : an important aera in the analysis and design of information system; in Management Informatics, Vol 3, n° 1, 1974; pg 45-54.

#### Isdos.-

- LESUISSE; Projet ISDOS; in Bulletin de liaison du Club de Base de données; n° 16, juin 1976; pg 166, 200.
- TEICHROEW; PSA : requirements for the problem statement analyzer; in COUGER, KNAPP; System Analysis techniques; Wiley 1974; pg 336, 357.
- NUNAMAKER; A methodology for the design and optimization of information processing systems; in AFIPS Conference proce-



- NUNAMAKER, KONSYSKI; From Problem Statement to automatic Code Generation; in systemeering 75 by LUNDEBERG; Student-litteratur Lund 1976.
- TEICHROEW, SAYANI; Automation of System building; in Data-mation, août 15, 1971, pg 25,29.

Autres systèmes :

- BOURGEOIS; Description fonctionnelle de l'organisation; Référence incomplète 1977.
- GOLVERS; Strutured Analysis and design of information systems; Référence incomplète 1976.
- IBM; SOP : Study organisation plan documentation techniques; in COUGER, KNAPP; System analysis techniques; Wiley 1974, pg 94,127.
- ORDEN; Information structure modeling of organisation; in Information processing 71; North-Holland Co 1972; pg 924-930.
- CGI; PAC (génération automatique et documentation); Compagnie Générale d'Informatique, Paris 1977.

Autres informations :

- BLUMENTHAL; MIS, a framework for planning and development; Prentice Hall 1969.
- BUKENKO, BERILD; From information requirements to DBTG-data Structure ; in Proceedings of ACM, 1976, Salt Lake City : Sigmod - Sigplan Conference.
- BURCH-STARTER; Information system : theory and practice ; Hamilton, Californie, 1974.
- DAVIS; MIS : Conceptual fundation, structure, development; McGraw Hill, 1975.
- EMERY; Organisational planning and control systems : Theory and Technology; Mc Millan 1971.
- GRAY, LONDON; Documentation Standards; Business book Ltd, London 1969.
- FNDP; Data structure models for information systems; Travaux de l'Institut d'informatique n° 4, PUNamur, 1975.
- KELLY; Computerized Management information system; Mc Millan 1970.
- LANGEFORS; Theoritical Analysis of information system; Student-litteratur Lund, 1974.
- LANGEFORS; Some approaches to the theory of information systems; in BIT 3 (1963); pg 220, 254.
- LANGEFORS; Information design computations using generalized matrix algebra; in BIT 5, 1965; p 96, 121.
- LANGEFORS, SUNDGREN; Information system architecture; Petrocelli/Charter; New York 1975.
- LEMOIGNE; Les systèmes de décision dans les organisations; PUF 1975.



- MARTZLOFF; Découvrir les systèmes; Ed. d'Organisation; Paris 1975.
- McDONOUGH; GARETT; Management systems; IZWIN, Homewood, Illinois, 1965.
- MURDICH-ROSS; Information Systems for modern management; Prentice Hall, Englewood Cliffs, New York, 1975.
- Stevens, Myers, Constantine; Structured Design; IBM System journal, Vol 13, n° 2, 1974.
- LANGEFORS, BUKENKO; Computer-aided Information system analysis and design; Student Litteratur Lund 1973.
- LUNDEBERG, SOLVBERG; Systemeering 75; Student Litteratur Lund 1976.
- COUGER, KNAPP; System Analysis Techniques; WILEY 1974.